Service-orientierte Architekturen

Leitfaden und Nachschlagewerk
2. Auflage
Neu enthalten: Fokusthema „SOA und Security“

BITKOM

Impressum

Herausgeber: BITKOM
Bundesverband Informationswirtschaft,
Telekommunikation und neue Medien e. V.
Albrechtstraße 10 A
10117 Berlin-Mitte
Tel.: 030.27576-0
Fax: 030.27576-400
bitkom@bitkom.org
www.bitkom.org

Ansprechpartner: Stephan Ziegler
Tel.: 030.27576-243
s.ziegler@bitkom.org

Redaktion: Stephan Ziegler, Anne Müller (BITKOM)

Gestaltung / Layout: Design Bureau kokliko / Anna Müller-Rosenberger (BITKOM)

Service-orientierte Architekturen – Leitfaden und Nachschlagewerk (2. Auflage)
Diese Auflage wurde um den Artikel „Strategie und Taktik“ und das Kapitel „SOA und Security“ ergänzt. Ansonsten erscheint der Leitfaden weitestgehend ohne wesentliche Änderungen zur vorherigen Auflage.

Service-orientierte Architekturen

Leitfaden und Nachschlagewerk

2. Auflage

Neu enthalten: Fokusthema „SOA und Security“

BITKOM

Inhaltsverzeichnis

Vorwort

Das vorliegende Dokument zu Service-orientierten Architekturen ist das Produkt der freiwilligen Zusammenarbeit von über 30 Unternehmen im BITKOM-Arbeitskreis SOA-Technologies. Der Inhalt entspricht weitestgehend dem Konsens, der in zahlreichen Diskussionen zwischen den beteiligten Softwareanbietern, IT-Beratungsunternehmen, IT-Service-Providern und Forschungsinstituten erarbeitet wurde.

Aufgrund der schnellen Entwicklung im Bereich der IT-Architekturen – insbesondere im SOA-Umfeld – stellen die verschiedenen Abschnitte lediglich den aktuellen Stand der Technik und verschiedener Methoden und Vorgehensweisen dar. Um die vorliegenden Abschnitte an die schnellen Veränderungen anpassen und ggf. ergänzen zu können, wurde bewusst eine zeitgemäße Form der Verbreitung für das gesammelte Wissen gewählt. Die gemeinsam durch Sponsoren finanzierte Plattform www.SOA-Know-How.de bietet neben der jeweils aktuellsten Version des vorliegenden Leitfadens Raum für Diskussionen zu den verfügbaren Artikeln. Nutzer haben die Möglichkeit, ihr Wissen und ihre Erfahrungen rund um SOA mit anderen Spezialisten zu erörtern und mit interessierten Anwendern zu teilen.

1 Einführung

1.1 Zielsetzung und Motivation des Leitfadens

Bei großen Unternehmen ist momentan der Trend zu beobachten, die Unternehmens-IT in Richtung flexibel konfigurierbarer, Service-orientierter Systeme umzugestalten. Dies hat im Wesentlichen zwei Ursachen: Ein Grund ist der Wunsch, Geschäftsprozesse möglichst flexibel an sich ändernde Gegebenheiten anpassen zu können. Klassische IT-Architekturen wie Client-Server- oder Mainframe-Systeme bilden in der Regel Geschäftsprozesse statisch ab und erlauben nur in einem sehr engen Rahmen Änderungen der durch sie unterstützten Geschäftsprozesse. Die inadäquate Darstellung der Prozesse durch die IT beschränkt damit die unternehmerischen Aktivitäten.

Der zweite Grund für den Wechsel hin zu Service-orientierten IT-Systemen liegt in den Kosten für die Wartung und Weiterentwicklung der IT-Infrastruktur eines Unternehmens. Bis Anfang der 90er-Jahre war es noch üblich, die IT in einem großen Unternehmen im Durchschnitt alle 7 Jahre vollständig auszutauschen, um den sich ändernden Anforderungen gerecht zu werden. Diese Möglichkeit nimmt aufgrund steigender Kosten stetig ab. Paul Strassmann war als CIO und in ähnlichen Positionen in multinationalen Konzernen und US-Regierungsbehörden tätig. Er berichtete 2001, dass unter seiner Verantwortung der erste Austausch der Unternehmens-IT im Jahr 1946 etwa 10 % des Investitionsvolumens betrug. Die letzte Umstellung Mitte der 90er-Jahre belief sich bereits auf 54 % des Investitionsvolumens. Ein weiterer Austausch wäre nicht mehr finanzierbar gewesen. Es wurden daher Wege gesucht, die Unternehmens-IT graduell anzupassen und weiterzuentwickeln, ohne sie komplett zu ersetzen.

Service-orientierte Architekturen (SOA) besitzen das Potenzial, Geschäftsprozesse und IT-Prozesse zu harmonisieren. Sie erlauben eine graduelle Weiterentwicklung der IT-Architektur. Da die Anforderung, flexibel und effizient auf sich ändernde geschäftliche Rahmenbedingungen reagieren zu können, in allen Domänen existiert, haben sich erstmals in der Geschichte die IT Hersteller auf ein Architekturparadigma und ein Konzept zur Interaktion von Services geeinigt. So können Services verschiedener Hersteller über Plattformgrenzen hinweg miteinander interagieren.

Zunächst fand die SOA-Debatte vorwiegend in der IT-Sphäre statt, mittlerweile treten deren Implikationen für die Unternehmensstrategie immer deutlicher zutage. Service-orientierte Architekturen versprechen eine engere Verzahnung der betriebswirtschaftlichen Bereiche mit der IT. Für die Geschäftsführung und das Management zeichnen sich neue Formen einer IT-gestützten Unternehmensführung ab. Hierfür ist ein integriertes Entwicklungs-, Anwendungs- und Infrastrukturmodell notwendig, welches Firmen bei der ganzheitlichen Analyse, der Definition von strategischen und betriebswirtschaftlichen Zielen sowie deren Umsetzung in Form von Softwarelösungen und Geschäftsabläufen unterstützt.

Service-orientierte Architekturen setzen so eine Zäsur in der Entwicklung von Softwaresystemen, die durch proprietäre Schnittstellen, monolithische Anwendungsarchitekturen und starre Geschäftsprozesse gekennzeichnet waren. Im Zentrum der Service-Idee stehen plattformunabhängige Bausteine, die sich mit klar definierten Schnittstellen beliebig zu lose gekoppelten Anwendungen oder Geschäftsprozessen kombinieren lassen. Angesichts der aktuellen Herausforderungen für Unternehmen erscheint ein solcher Ansatz zunehmend attraktiv: Regulatorische Anforderungen wie Basel II und Sarbanes-Oxley, Risiko-Management oder Performance Management erfordern neue Prozesse. Darüber hinaus sind Unternehmen einem steigenden Rationalisierungsdruck und schärferen Wettbewerbsbedingungen ausgesetzt und müssen darauf mit flexibleren, automatisierten Abläufen schnell und kosteneffektiv reagieren.

Ziel des vorliegenden Leitfadens ist es, die Vorteile und Herausforderungen beim Einsatz einer SOA im

Unternehmen kritisch und herstellerunabhängig zu beleuchten.

1.2 Organisation und Aufbau des Leitfadens

Der Leitfaden will bei der Annäherung an das so vielschichtige und komplexe Thema der Service-orientierten Architekturen unterstützen und zur Versachlichung der Diskussion rund um SOA motivieren.

Die vorliegende Orientierungshilfe enthält eine umfangreiche Sammlung von Artikeln, die ständig weiterentwickelt und ergänzt werden. Die aktuellsten Versionen sind online unter www.SOA-Know-How.de kostenlos verfügbar und wurden anhand von Themenkomplexen entlang eines SOA-Lebenszyklus gruppiert.

Im Kapitel „Einführung" wird neben einer Beschreibung der Ziele des Leitfadens der Versuch unternommen, eine Definition für Service-orientierte Architekturen zu formulieren, die den verschiedenen Sichtweisen auf das Thema gerecht wird. Darüber hinaus werden hier grundlegende Fragestelllungen, wie beispielsweise die Bedeutung von SOA für die gesamte IT-Industrie erörtert.

Anschießend wird im Kapitel „Warum SOA?" der Frage nachgegangen, warum ein so tief greifender Umbau der bestehenden Anwendungslandschaften überhaupt notwendig ist und wo die Grenzen bisheriger System-Architekturen liegen. Darüber hinaus werden Potenziale und Risiken von Service-orientierten Architekturen erörtert. Es wird verdeutlicht, dass SOA weit mehr ist als eine Informationstechnologie und welche unterschiedlichen Bereiche einer Organisation bzw. eines Unternehmens durch die Umsetzung von Service-orientierten Architekturen berührt werden.

Mittlerweile hat sich herumgesprochen, dass es SOA nicht „von der Stange" gibt. Aber wie kommt man dann zu einer Service-orientierten Architektur? Das Kapitel „Vorbereitung und Planung" beschreibt fundamentale Voraussetzungen und stellt eine Orientierungshilfe für die ersten Schritte in Richtung Serviceorientierung dar.

Im Kapitel „Architektur und Aufbau" wird anhand einer Referenzarchitektur erklärt, welche Elemente eine vollständig ausgebaute Service-orientierte Architektur enthalten sollte und welche Aufgaben die einzelnen Elemente übernehmen. Mögliche Vorgehensweisen zur Verzahnung von IT- und Fachabteilungen und die damit verbundene Abbildung von Geschäftsprozessen in der IT-Landschaft werden vorgestellt und die Granularität von Services diskutiert.

Der Betrieb von Service-orientierten Architekturen unterscheidet sich in wesentlichen Punkten von bisherigen Betriebskonzepten. IT-Prozesse wie beispielsweise Architekturmanagement oder die Überwachung von Diensten bzw. Services weichen von Verfahren konventioneller Client/Server-Architekturen ab. Neben dem Thema der neuen Betriebsführung finden sich im Kapitel „Umsetzung und Betrieb" auch Informationen zu SOA-Governance. Der Frage nach dem Zusammenhang zwischen Service-orientierten Architekturen und dem Betriebsmodell hinter Software as a Service (SaaS) wird hier zusätzlich nachgegangen.

Gezielte Hilfestellung bei der Erarbeitung von geeigneten Konzepten im Bereich der IT-Sicherheit bietet das Kapitel „SOA und Security". Sicherheitsanforderungen werden definiert und Lösungsmuster für Security-Implementierungen vorgestellt. Letztlich wird deutlich: Durch intensive Auseinandersetzung und richtige Wahl der Methoden kann ausgeschlossen werden, dass Service-orientierte Architekturen – wie oft behauptet – zum Sicherheitsrisiko werden.

Im Kapitel „Ausblicke" wird abschließend dargestellt, warum Service-orientierte Architekturen kein kurzfristiger IT-Trend sind und welche anderen Technologien Einfluss auf mögliche Entwicklungen rund um SOA haben können. Ergänzend gibt das Kapitel „Standards" einen Überblick zu Normen, die beim Aufbau einer Service-orientierten Architektur Anwendung finden können. Die Notwendig-

keit von Standards zur semantischen Beschreibung von Diensten bzw. Services wird diskutiert.

HINWEIS: Unter www.SOA-Know-How.de ist der vorliegenden Leitfaden online – ggf. in Teilen auch in aktuellerer Auflage – verfügbar. Interessierte Leser sind eingeladen, auf der Web-Plattform die einzelnen Abschnitte zu bewerten und durch die Beteiligung an Diskussionen im Forum an der ständigen Verbesserung und Aktualisierung mitzuwirken.

1.3 SOA: Definition und Abgrenzungen

Vor etwa fünf Jahren wurde das Thema service-orienterte Architekturen hauptsächlich auf technischer Ebene diskutiert. Heute dagegen wird der Begriff inflationär verwendet – weit über die ursprüngliche Bedeutung im Sinne eines IT-Architektur-Konzeptes hinaus. Nicht erst seit dem für IT-Unternehmen mitunter leidvollen Beginn des einundzwanzigsten Jahrhunderts ist klar: Ein neues Konzept lässt sich nur mit glaubhaften, ökonomisch sinnvollen Mehrwertdarstellungen erfolgreich am Markt etablieren. Die entscheidenden Argumente für die Einführung einer Service-orientierten Architektur in einem Unternehmen lassen sich direkt aus den Bedürfnissen von Anwendern und Management ableiten. Letztere betrachten die IT mit Recht lediglich als ein Werkzeug zur Durchsetzung von geschäftlichen Interessen. Mit anderen Worten: Wertschöpfungsketten als Kern von Unternehmen und als Grundlage unternehmerischen Handelns müssen möglichst optimal durch die eingesetzte Informationstechnologie unterstützt werden. Der Erfolg von Wertschöpfungsketten und Geschäftsmodellen ist in vielen Branchen, wie bei Banken und Versicherungen sowie in der Telekommunikations-, Automobilindustrie- und Logistikindustrie, zunehmend von der IT-Unterstützung abhängig. In einigen Branchen sind die heutigen Geschäftsprozesse erst durch Informationstechnologie möglich geworden.

Aufgrund von Globalisierung und weltumspannender Wertschöpfung sind Unternehmen gezwungen, dem zunehmenden Druck des Marktes zu begegnen. Gefragt ist in erster Linie unternehmerische Wettbewerbsfähigkeit durch Flexibilität, d. h., Prozesse, Geschäftsdaten sowie Organisationsstrukturen müssen effektiv den jeweiligen Gegebenheiten auf dem Markt angepasst werden. Je größer der IT-Anteil an der Wertschöpfung, umso flexibler bzw. agiler müssen folglich auch die IT-Systeme und Anwendung sein. Das zentrale geschäftspolitische Ziel des Einsatzes Service-orientierter Architekturen ist demnach eine effektive Umsetzung von Anforderungen, die sich aus flexiblen Geschäftsprozessen ableiten. Im Gegensatz zu bisherigen Ansätzen muss sich das Werkzeug „Informationstechnik“ den geschäftlichen Bedürfnissen unterordnen. Damit verändert sich das Zusammenspiel von Geschäftsmodell (Prozesse, Organisation und Geschäftsobjekte) und IT. Eine SOA verursacht bei dieser auch als Business-IT-Alignment bezeichneten Verzahnung Anpassungsbedarf auf beiden Seiten. Die Umsetzung des Konzeptes SOA bedingt neben dem technischen Umbau der bestehenden IT-Architektur insbesondere einen neuen Umgang mit IT durch das Management und die Fachabteilungen. Diese Erkenntnis ist der Grund, warum SOA in den letzten Jahren zunehmend ein Management-Thema geworden ist. Die bisher vorherrschenden großen aber unflexiblen Anwendungen sollen in leicht anpassbare Software-Einheiten überführt werden, die jeweils eine definierte Funktionalität als Service bereitstellen, sodass Teilprozesse der Wertschöpfungskette durch Kombination dieser Software-Einheiten (Services) optimal unterstützt werden können. Dabei spielt für Management und Fachabteilung die technische Umsetzung immer weniger eine Rolle. Die schnelle Verfügbarkeit von möglichst leicht in die Wertschöpfungskette zu integrierenden Services tritt in den Vordergrund.

Blendet man bei der Betrachtung von SOA jegliche Technologie vorübergehend aus, so lässt sich eine beliebige Wertschöpfungskette als Aneinanderreihung von aufeinander abgestimmten Dienstleistungen (Services) oder auch als Folge der Nutzung von Services betrachten. Jeder einzelne Service trägt direkt bzw. indirekt zur Wertschöpfung bei. Aus Sicht des Managements steht dabei eine flexible und in jeglicher Hinsicht effiziente Leistungserbringung im Mittelpunkt. Geeignete Regeln, Prozesse und eine auf die Wertschöpfung durch Services ausgerichtete Organisation sind dafür notwendig. Neben der flexiblen

Anpassung der Wertschöpfungsprozesse bietet SOA einen weiteren Vorteil, denn auf die Services kann aus allen Wertschöpfungsprozessen heraus zugegriffen werden. Somit können häufig verwendete Funktionalitäten einmal realisiert und mehrfach verwendet werden. Früher wurden gleiche Funktionalitäten für jede Anwendung separat entwickelt.

Services können teilweise oder ganz durch Informationstechnologie erbracht werden. Folglich müssen Anwendungen und Systeme, welche Services zur Verfügung stellen, so konstruiert sein, dass sie möglichst kostengünstig und schnell an neue Anforderungen angepasst werden können. Ein vielversprechender Ansatz für flexible Anwendungssoftware sind Service-orientierte IT-Architekturen. Es handelt sich dabei nicht um eine neue Technologie und vor allem nicht um ein dediziertes Produkt eines Unternehmens. Ihr Kernstück bilden Services, die unabhängig von Anwendungen und Plattformen sind. Sie kapseln komplexe Anwendungslogik, welche durch eine Menge von Systemen bzw. Softwarekomponenten angeboten werden. Sie sind technikneutral, selbstbeschreibend, lassen sich miteinander beliebig in einer standardisierten Form kombinieren, regelbasiert steuern und wiederverwenden. Im Resultat entsteht eine modularisierte und flexible Anwendungslandschaft: Diese ist Grundlage für Flexibilität bzw. Agilität und Produktivität im Geschäftsleben, sie ist auch Grundlage für die Verbesserung von Effektivität und Qualität der IT-gestützten Prozesse vor dem Hintergrund einer sich permanent erhöhenden Marktdynamik.

Aus der allgemeinen Definition für SOA im Sinne eines unternehmerischen Prinzips kann eine weitere konkretere Definition bzgl. einer Service-orientierten IT-Achitektur abgeleitet werden.

In der Literatur werden für jede der in den vorangegangenen Absätzen beschriebenen Sichtweisen auf SOA – wertschöpfungsorienterte oder IT-Architekturorienterte Sicht – unterschiedliche Definitionen formuliert. Dieser Ansatz ist aber mitunter hinderlich bei der konkreten Umsetzung, weil das Verständnis der Ziele und Herausforderungen der jeweils anderen Seite enorm wichtig ist. Beispielsweise muss die IT-Seite den Ansatz der Wertschöpfung durch Services verinnerlicht haben.

Definition

Eine Service-orientierte Architektur (SOA) ist ein Konzept, welches das Geschäft und die IT eines Unternehmens nach Diensten strukturiert, die modular aufgebaut sind und flexibel zur Umsetzung von Geschäftsprozessen kombiniert werden können.
Die IT-spezifischen Bestandteile des Konzeptes werden durch eine IT- bzw. System-Architektur umgesetzt, welche auf lose gekoppelten und technisch voneinander unabhängigen Services beruht, deren Interoperabilität auf offenen Standards basiert.

Fazit:

- Eine Service-orientierte Architektur lässt sich nicht fertig auf dem Markt kaufen.
- Vielmehr ist SOA ein Konzept, das jeweils auf die individuellen Gegebenheiten – sowohl hinsichtlich der Organisation als auch der bestehenden Anwendungslandschaft – eines Unternehmens angepasst werden muss.
- Eine Service-orientierte Architektur ist nie abgeschlossen, sondern entwickelt sich mit dem Unternehmen vor dem Hintergrund eines dynamischen Marktumfeldes immer weiter.

1.4 Für wen ist die Service-orientierte Architektur relevant?

Diese Frage betrifft zwei Bereiche: Zum einen geht es darum zu klären, für welche Unternehmen eine mögliche SOA-Initiative in Betracht kommt. Kriterien und Ansatzpunkte sind gefragt, mit deren Hilfe sich abschätzen lässt, ob eine Investition in SOA für ein Unternehmen sinnvoll bzw. notwendig ist. Diese Kriterien stellen nur einen ersten Ansatzpunkt dar, denn eine Beurteilung der Relevanz von SOA für ein Unternehmen kann nur im konkreten Unternehmenskontext getroffen werden.

Im ersten Schritt wird darüber Auskunft gegeben, wer zu den potenziellen Anwendern gehört. Der zweite Teil beschäftigt sich mit einer anderen involvierten Zielgruppe, für die eine Auseinandersetzung mit dem Thema SOA notwendig ist. Es gilt zu klären, welche Personen innerhalb der Unternehmen an der Planung und Umsetzung einer SOA-Initiative beteiligt werden sollten.

Für welche Unternehmen ist SOA relevant?

Im Kontext dieses Dokuments soll die Frage beantwortet werden, ob SOA auch für kleine und mittelständische Unternehmen eine Relevanz hat. Oder noch deutlicher „komme ich als Vertreter des Mittelstandes um das Thema SOA herum?“. Während Großunternehmen und Konzerne die finanzielle Stärke haben, auch umfangreichere SOA-Initiativen zu starten, stehen kleinere Unternehmen bezüglich der Kosten und kurzer Return on Investment (ROI)-Zyklen unter stärkerem finanziellen Druck. Ebenso fehlt oft die nötige Personalstärke, um das entsprechende fachliche Know-how aufzubauen und die notwendigen Rollen im damit verbundenen Veränderungsprozess zu besetzen. Hinter der Frage steckt demnach eher die Hoffnung, dass es Rahmenbedingungen gibt, die einen Verzicht auf ein hohes Investment und Risiko für den Umbau der Softwarelandschaft in eine Service-orientierte Architektur ohne Nachteile für das Unternehmen ermöglichen. Ist die Relevanz von SOA also von der Unternehmensgröße abhängig? Die Frage kann mit einem klaren „Nein“ beantwortet werden.

Es versteht sich von selbst, dass nicht jede Problemstellung im Unternehmen mit SOA gelöst werden kann, auch wenn dies auf dem Höhepunkt der medialen Lobpreisungen oft vermittelt wurde. Wie der Nutzen und die Notwendigkeit kann die Relevanz nur im jeweiligen Unternehmenskontext bewertet werden. Es gibt jedoch einige Kriterien und Anhaltspunkte, die eine Auseinandersetzung mit SOA besonders angebracht erscheinen lassen.

Die wesentlichen Kriterien lassen sich in die Bereiche Markt, Unternehmen und IT aufteilen. Eine gute Übersicht darüber findet man z. B. im Blog von Thomas Müller, der die folgenden Kriterien identifiziert: [MUE 2007]

Markt

Service-orientierte Architekturen sind insbesondere für solche Unternehmen interessant, die in einem Markt operieren, der sich durch Dynamik, Änderungs- und Anpassungsfähigkeit auszeichnet. Dies können z. B. Märkte sein, die häufig gesetzliche Änderungen umsetzen müssen oder von einem hohen Wettbewerbsdruck geprägt sind. Ein weiteres Kennzeichen solcher Märkte ist die notwendige hohe Reaktionsgeschwindigkeit in Bezug auf Preise, Produkte und Dienstleistungsangebote.

Dies gilt insbesondere für Märkte, in denen der Einfluss von Internet und B2B-Szenarien neue Geschäftsmodelle ermöglicht bzw. erzwingt. So werden bestimmte Versicherungsangebote heute zunehmend online verglichen und abgeschlossen. Prozesse werden stärker automatisiert und direkt von Endkunden ausgelöst (z. B. Onlinebanking). Produkte werden über Online-Auktionen oder Online-Marktplätze veräußert.

Unternehmen

Bisher wurden viele Dienstleistungen über eine persönliche Kommunikation – also Mensch-zu-Mensch – oder den Austausch von gedruckten Geschäftsunterlagen angeboten. Für derartige Abwicklungen blieb die IT im Hintergrund. Zunehmend jedoch werden Dienstleistungen über eine reine Maschinenkommunikation durchgeführt. Dazu müssen Services, die bisher nur unternehmensintern relevant waren, über Unternehmensgrenzen hinweg bereitgestellt werden. Aus Sicht der Unternehmen ist SOA also dann relevant, wenn an der Abwicklung von

Unternehmensprozessen Partner und Kunden beteiligt sind. Gleiches gilt für den Fall, dass Zulieferer mit ihrer IT stärker in die Wertschöpfungsketten ihrer Auftraggeber eingebunden werden, um die Durchlaufzeiten zu optimieren. Aber es entstehen auch neue Dienstleister, die hauptsächlich vorhandene Dienstleistungen kombinieren und ergänzen.

Ein weiterer Ansatzpunkt für eine sinnvolle Auseinandersetzung mit SOA ist ein starkes (geplantes) Unternehmenswachstum; insbesondere dann, wenn dieses Wachstum durch Zukäufe von Unternehmen erfolgt. Durch Unternehmensaufkäufe und -zusammenschlüsse entstehen häufig funktional redundante Softwarelandschaften, oft auf Basis unterschiedlicher Technologien. In einem Unternehmen, in dem sich die IT durch fachlich orientierte Services auszeichnet, die sich bei Bedarf neu kombinieren lassen, kann eine notwendige Konsolidierung oder ein Wachstum der IT wesentlich schneller und kostengünstiger durchgeführt werden, indem neu hinzukommende Services in die vorhandenen Geschäftsprozesse integriert werden.

IT

Unternehmensstrategie und Geschäftsmodell bilden die Grundlage für SOA. In einigen wichtigen Punkten müssen jedoch auch Aspekte aus dem Bereich der IT berücksichtig werden.

SOA ist vor allem dann relevant, wenn ein Unternehmen und dessen Geschäft maßgeblich von der IT abhängen. Ist dies nicht der Fall, dürften Investitionen in eine SOA kaum durchsetzbar und sinnvoll sein. Dies trifft hauptsächlich auf die bereits erwähnten Kleinunternehmen zu. Bei mittelständischen Unternehmen muss individuell untersucht werden, inwieweit Änderungen der IT das Geschäft positiv beeinflussen können. Bei Großunternehmen stellt sich diese Frage in der Regel nicht. Werden die Geschäftsprozesse maßgeblich durch die IT unterstützt, ist die Beschäftigung mit SOA obligatorisch, dies umso stärker, wenn die IT darüber hinaus strategische Bedeutung für das Unternehmen hat.

Auf der IT-Ebene ist SOA besonders dann relevant, wenn eine sehr heterogene IT-Landschaft existiert und Anwendungen auf Basis unterschiedlicher Technologien miteinander kommunizieren müssen. Liegen solche Rahmenbedingungen im Unternehmen vor, hat SOA einen hohen Nutzen.

Für die meisten Unternehmen wird die Auseinandersetzung mit dem Thema SOA aktuell, wenn Modernisierungsprojekte oder Neuentwicklungen anstehen und die Chance für eine Neuausrichtung der IT oder Teilen davon besteht. Diese Neuerungen können unterschiedliche Ausprägungen annehmen:

- Ablösung bestehender Alt-Systeme durch Kaufprodukte oder Eigenentwicklungen
- Erweiterung der Nutzen-Szenarien vorhandener Software (Service-Enablement)
- Erweiterung vorhandener Software um zusätzliche (individuelle) Funktionen
- Neue Systeme

Für die Optimierung heterogener Systemlandschaften bietet SOA interessante Möglichkeiten. Ziel ist letztlich die Optimierung des gesamten Systems, das Konzept SOA lässt jedoch die Umstellung einzelner Teilbereiche der IT in kleinen Schritten zu und bietet damit Investitionssicherheit. Davon profitieren insbesondere mittelständische Unternehmen mit eingeschränkten Budgets und kurzem ROI (siehe auch Kapitel „Erfolgsfaktoren“).

Die direkte Unterstützung betriebswirtschaftlicher Abläufe durch Software-Services, die wiederum Prozessketten im Sinne der Wertschöpfung durchlaufen, macht das Besondere des SOA-Konzeptes aus. Ändern sich Geschäftsabläufe sowohl innerhalb des Unternehmens als auch in globalen Wertschöpfungsnetzen, so lassen sich die begleitenden Software-Komponenten im Baukastenprinzip mit vergleichsweise einfachen Mitteln neu zusammenstellen. So wird es Unternehmen jeglicher Größe möglich, Flexibilität und Wirtschaftlichkeit zugleich zu steigern. Auch wird sich in wenigen Jahren Betriebswirtschaftliche Software auf Basis von SOA im Bereich der Unternehmenssoftware durchsetzen.

Beteiligung von Unternehmensleitung, Fachbereichsleitung und IT-Management

Voraussetzung für solche Änderungen ist allerdings, dass Unternehmensleitung, Fachbereichsleitung und die IT-Verantwortlichen die eigenen Unternehmensprozesse durchdrungen haben und sich ihrer Kernkompetenzen bewusst sind. Dieser Aspekt benennt eine wichtige Problematik im Kontext Service-orientierter Architekturen. Erst das tief gehende Verständnis der betriebswirtschaftlichen Abläufe im Unternehmen sowie der Differenzierung von dem Wettbewerb ermöglichen es, stabile Service-orientierte Software-Services zu entwickeln, um genau diese Geschäftsabläufe zu optimieren. Bei vielen Unternehmen herrscht hier Nachholbedarf. Um optimale Analyseergebnisse zu erhalten, ist ein gemeinsames Zusammenarbeiten zwischen Fachbereichen und IT erforderlich, da ein wesentlicher Anteil des Erfolges von SOA darin liegt, dass Fachbereiche und IT ineinandergreifen. SOA-Projekte ohne Mitwirkung beider Interessengruppen nutzen das Potenzial von SOA nicht bzw. nur ungenügend. Erst das Zusammenspiel aller Interessengruppen bei Planung und Umsetzung der SOA-Strategie stellt sicher, dass die Anforderung berücksichtigt werden und das Potenzial so ausgeschöpft werden kann.

SOA: Relevant für alle in der IT-Industrie

In diesem Zusammenhang werden zukünftig andere Fragen als die der Relevanz in den Vordergrund rücken: „Von wo und wem erhalten Unternehmen zukünftig Software-Lösungen? Wie werden die Leistungen abgerechnet und verwaltet. Wo liegen die Informationen und wie sind sie abgesichert?“ Ein Geschäftsführer oder eine Unternehmensführung haben selten das IT-Know-how für die Entwicklung eines eigenen SOA-Konzeptes. Hier wird oft auf qualifizierte Beratung aus IT-Systemhäusern zurückgegriffen, die bei der Einführung einer SOA-basierten Geschäftslösung unterstützen. Gerade für mittelständische Systemhäuser hält das SOA-Konzept interessante Geschäftschancen bereit: IT-Dienstleister entwickeln aus einzelnen Komponenten branchenspezifische oder spezialisierte Anwendungskomponenten und integrieren diese künftig in die Geschäftsprozesse des Kunden. Somit spielen IT-Systemhäuser und Softwareprovider eine wichtige Rolle bei der Implementierung von SOA-Konzepten. Erst durch ihre intensive Beratungsleistung lässt sich aus dem „SOA-Konzept“ ein echter Mehrwert für mittelständische Unternehmen generieren. IT-Partner wie IT-Beratungshäuser oder IT-Dienstleister wissen, mit welchen Infrastrukturen sie Lösungen schaffen können, die die Unternehmen ihrer Kunden zukunftsträchtig machen.

Zukünftig wird aufgrund der technologischen Entwicklung um SOA kein Weg herumführen. Wie sich die Verzahnung der betriebswirtschaftlichen Bereiche eines Unternehmens mit der IT auf technologischer Ebene gestaltet, ist für den einen oder anderen Unternehmer von sekundärer Bedeutung. Entscheidend ist, dass SOA die Wirtschaftlichkeit seines Unternehmens erhöht und dessen Wettbewerbsfähigkeit steigert. Dabei gewinnt die Einbindung der Konzepte „Web 2.0“ und „Software as a Service“ (SaaS) eine weitere wichtige Bedeutung für die Entwicklung zusammengesetzter Geschäftsarchitekturen. Hosted Services, Software-Services oder Software as a Service (SaaS) entwickeln sich derzeit zu einem Trend. Hinter diesen Begriffen steckt die Idee: Software wird nicht mehr gekauft und auf den Computern im Unternehmen installiert, sondern die Programme laufen auf dem Server eines Dienstleisters, der sie wiederum via Internet einer Firma zur Verfügung stellt.

Für die IT-Industrie wird SOA zum führenden Paradigma. Anbieter von Infrastruktursoftware entwickeln ihre Produkte zu SOA-Plattformen weiter. Softwarehersteller machen ihre Anwendungssoftware SOA-fähig, um am Markt bestehen zu können. Neben Softwareherstellern etablieren sich Serviceanbieter, die die Services sicher verfügbar auf Basis von Service Level Agreements betreiben. Unternehmen, die sich mit SOA beschäftigen, benötigen Berater, die bereits Erfahrung mit SOA haben. Für Systemanalytiker reicht die Betrachtung von Anwendungen und Use Cases aus Sicht einer einzelnen Fachabteilung nicht mehr aus. In einem SOA-Umfeld rücken bei der Systemanalyse die Betrachtung von übergreifenden Prozessen

und ganzen Systemlandschaften sowie die Integrationsfähigkeit einzelner Services in eine solche Landschaft in den Fokus. Schließlich müssen sich auch Programmierer zunehmend mit Service-orientierten Prinzipien sowie entsprechenden Technologien und Standards auseinander setzen. Obwohl SOA für Programmierer hauptsächlich eine Kombination bewährter Prinzipien darstellt, kommt die Auswirkung einem Paradigmenwechsel gleich – wie ehemals von der strukturierten zur objektorientierten Programmierung.

Dies stellt auch neue Anforderungen an die Ausbildung an den Hochschulen. Sowohl in der betriebswirtschaftlichen als auch in der informationstechnischen Ausbildung müssen die Konzepte und Prinzipien von SOA vermittelt werden. Damit werden die Voraussetzungen geschaffen, dass neue Mitarbeiter das Gedankengut bereits mit in die Unternehmen bringen. So wird zukünftig die Zusammenarbeit von Business und IT auf Basis eines gemeinsamen Verständnisses und einer gemeinsamen Sprache von Grund auf unterstützt.

Fazit

- Für fast alle Unternehmen (abgesehen von einigen Kleinunternehmen) ist es sinnvoll, sich generell mit der Frage einer Anpassung von Geschäftsprozessen und IT-Strukturen im Sinne von SOA zu beschäftigen. Ob sich eine Investition letztlich lohnt bzw. notwendig ist und welches Ausmaß die SOA-Initiative haben wird, ist dann individuell zu klären.
- Innerhalb von Unternehmen sollten sowohl Management als auch IT-Verantwortliche an der Frage einer möglichen Veränderung beteiligt werden. In die Planung und Umsetzung einer SOA-Initiative sollten darüber hinaus auch Mitarbeiter aus den Fachabteilungen einbezogen werden. Letztendlich soll gewährleistet sein, dass Geschäftsprozesse so gut wie möglich analysiert und optimiert werden.
- Für Software-Hersteller, Beratungsfirmen, Analysten etc. ist eine Auseinandersetzung mit dem Thema unverzichtbar.

Weiterführende Links

[MUE2007] „SOA im Mittelstand – Wie sinnvoll ist SOA?“, Thomas Müller, T-Systems SOA-Blog 18.04.2007 http://www.exploresoa.de/soa/de/soablog

[SON2007] „Wer braucht SOA?“, Karin Sondermann, SOA-Expertenrat, Computerwoche 03.08.2007, http://www.computerwoche.de/soa-expertenrat/?p=194

2 Warum SOA?

2.1 Nutzen und Notwendigkeit

Erhöhung der Agilität

Die Globalisierung der Märkte und eine wachsende Zahl ähnlicher Produkte zwingen Unternehmen zunehmend zur Differenzierung gegenüber den Mitbewerbern. Erfolg hängt zunehmend vom richtigen Geschäftsmodell ab. Agilität und Innovationskraft zählen zu den wichtigsten Fähigkeiten eines Unternehmens. Vor diesem Hintergrund benötigt ein erfolgreiches Management eine leistungsfähige, flexible IT, die alle Geschäftssysteme miteinander verknüpft. Eine IT, die auf der Basis von Serviceorientierung den rollenbasierten Zugriff auf sämtliche kontextspezifischen Informationen bietet und die Produktivität von Führungskräften sowie Mitarbeitern durch flexible Geschäftsabläufe verbessert. Viele Geschäftsprozesse sind heute ohne IT-Unterstützung nicht denkbar. Durch zunehmende Prozessautomatisierung werden noch nicht ausgeschöpfte Einsparpotenziale erschlossen. Eine funktionierende IT-Landschaft wird zum Herzschlag des Unternehmens. Die notwendige Beweglichkeit eines Unternehmens spiegelt sich im idealen Fall also in einer sich stetig entwickelnden IT-Landschaft wider. Krankt die IT, so ist auch das Unternehmen nicht mehr leistungsfähig.

Viele Unternehmen haben mittlerweile erkannt, dass die Weiterentwicklung ihrer Waren, Produkte und Dienstleistungen allein nicht mehr ausreicht, um im Wettbewerb zu bestehen. Was zählt, sind Geschwindigkeit und Wandlungsfähigkeit, denn nur wer sich schnell an veränderte Marktsituationen und Kundenansprüche anpassen kann, wird im Wettbewerb den entscheidenden Schritt voraus sein. Firmen wenden sich aus diesem Grund verstärkt der Innovation ihrer Geschäftsabläufe und den Service-orientierten Geschäftsmodellen zu. Das richtige Geschäftsmodell finden jedoch nur, wer sein Geschäft genau kennt, neue Business-Chancen rechtzeitig erkennt und diese kreativ und innovativ umsetzt, um so dem Wettbewerb einen Schritt voraus zu sein.

Doch notwendige Anpassungsfähigkeit verbunden mit kurzen Time-to-Market-Zyklen ist nichts Neues. Was sich ändert, ist die zunehmende Geschwindigkeit der Änderungen. Davon sind nicht nur unternehmensinterne Geschäftsprozesse betroffen, sondern zunehmend auch Geschäftsabläufe über Unternehmensgrenzen hinweg. Partnerunternehmen werden stärker in die eigene IT integriert und Outsourcing von IT-Dienstleistungen wird in einer Service-orientierten Architektur attraktiver. Durch diese Entwicklung bieten sich gerade für mittelständische Unternehmen völlig neue Marktchancen.

Viele historisch gewachsene monolithische Anwendungssysteme sind diesen Herausforderungen nicht gewachsen. Diese Systeme können jedoch nicht von heute auf morgen abgelöst werden. SOA als langfristiges Paradigma gewährleistet die Wandlungsfähigkeit der Unternehmens-IT durch die flexible Kombinationsmöglichkeit vorhandener Softwaredienste (Services) bei gleichzeitigem Investitionsschutz. Anwendungslandschaften werden ausgehend von den wertschöpfenden Geschäftsprozessen in sinnvolle Einheiten aufgeteilt, geordnet und bei Bedarf neu kombiniert. Dabei richtet sich die IT strikt an den Anforderungen des Geschäfts aus.

Auf dieser Grundlage wird die Voraussetzung für die Erhöhung der Agilität des Unternehmens geschaffen:

- Durch eine flexiblere Kombinationsmöglichkeit der Softwarebausteine können Geschäftsprozesse besser unterstützt werden.
- Neue wettbewerbsfähige Geschäftsmodelle können schneller auf den Markt gebracht werden.
- Neue dynamische Geschäftsmodelle (z. B. durch die Wahlmöglichkeit zwischen mehreren Anbietern für einen Service) werden möglich.
- Durch eine starke Ausrichtung der IT am Geschäft werden IT und Fachabteilung stärker zusammenwachsen (müssen) und damit werden so weniger Fehler an der Schnittstelle auftreten.

Senkung der Kosten

Obwohl die IT damit stärker in den Fokus der Unternehmensstrategie rückt, wird sie in der Regel immer noch als Kostenfaktor wahrgenommen. Die personalintensive Entwicklung und Pflege von Anwendungssoftware bei steigenden Personalkosten gerät immer wieder unter Druck: Forderungen nach Kostensenkung und Effizienzsteigerung bei hoher Qualität und schnellerer Time-to-Market werden an die IT-Abteilungen herangetragen. Nahezu alle Trends der Softwareentwicklung versuchten in der Vergangenheit mehr oder weniger erfolgreich diesen Forderungen gerecht zu werden. SOA – als eine Kombination bewährter Konzepte aus der IT-Branche – verfolgt dabei mehrere Richtungen:

- Doppelentwicklungen werden durch Wiederverwendung von Services vermieden.
- Es werden Voraussetzungen geschaffen, um IT-Services, die außerhalb des Kerngeschäfts liegen, günstiger zu beziehen und in die bestehende Anwendungslandschaft integrieren zu können.
- Entwicklungskosten werden durch einfachere Änderungen IT-gestützter Prozesse gesenkt.

Gerade zu diesen Punkten gehen die Meinungen der SOA-Experten auseinander. Während die Erhöhung der Agilität eines Unternehmens durch eine SOA für die meisten außer Frage steht, werden Kostenreduzierung und hoher Wiederverwendungsgrad infrage gestellt.

Neben Studien, die Kostenersparnisse nachweisen, verbreitet sich die Meinung, dass Service-orientierte Architekturen zunächst hohe Investitionen erfordern, die erst langfristig die IT-Kosten senken. Der ROI ergibt sich vor allem aus dem Geschäftsnutzen und den daraus resultierenden Umsatzzuwächsen.

Ähnlich verhält es sich mit der Wiederverwendung. Verwendungsfaktoren zwischen anfänglich 1 und im fortgeschrittenen Stadium bis zu 4 (d. h. ein Service wird im Schnitt von 1 bis 4 Konsumenten verwendet) sind realistische Werte.

SOA-Initiativen sind für jedes Unternehmen ein individuelles Vorhaben. Deshalb ist für die Beantwortung der Frage nach Notwendigkeit und Nutzen eine individuelle Betrachtung im jeweiligen Unternehmenskontext notwendig. Weitere Aspekte dazu finden sich im Kapitel „Für wen ist SOA relevant?“.

Fazit

Der Nutzen und die Notwendigkeit einer Service-orientierten Architektur ergibt sich aus der Flexibilisierung der Unternehmens-IT durch die Kombinationsmöglichkeit von Services und die Senkung der IT-Kosten durch Wiederverwendung vorhandener Services.

Weiterführende Links

[MUE2007] „SOA im Mittelstand – Wie sinnvoll ist SOA?“, Thomas Müller, T-Systems SOA-Blog 18.04.2007 http://www.exploresoa.de/soa/de/soablog

[SON2007] „Wer braucht SOA?“, Karin Sondermann, SOA-Expertenrat, Computerwoche 03.08.2007, http://www.computerwoche.de/soa-expertenrat/?p=194

[HER2006] „Wie sich SOA-Projekte rechnen“, Wolfgang Herrmann, Computerwoche 16.02.2006, http://www.computerwoche.de/soa-trends/572394/

Literatur

[BEN2006] „SOA erfolgreich nutzen“ Guido Bening, Andreas Zimmer, JavaSpektrum 3/2006

[STA2007] „SOA-Expertenwissen“, Gernot Starke u. Stefan Tilkov, dpunkt.verlag 2007

2.2 Problemstellungen und Lösungsansätze

In vielen Unternehmen ist die Kluft zwischen der IT und den Fachbereichen in den vergangenen Jahren stark gewachsen: Etablierte IT-Infrastrukturen ließen sich nur ungenügend an die sich immer rascher verändernden Anforderungen des Geschäftes anpassen. Folgende Situationen sind heute häufig anzutreffen:

Geschäft

- Erweiterung zu multinationaler, global tätiger Firma
- Geschäftsmodelle verändern sich im Wettbewerbsdruck rascher als zuvor
- Lieferanten und Abnehmer bewegen sich in immer zeitkritischeren Prozessketten
- Kunden sind zunehmend kritisch aufgrund weltweiten Wettbewerbs
- Unternehmensfusionen und Unternehmenskäufe
- Konkurrenzverhalten einzelner Abteilungen verschärfen die internen Probleme (Silodenken)

IT

- Verschiedene Systeme unterschiedlicher Qualität aus vergangenen Jahren, die nicht abgelöst werden können
- Gemessen am Budget zu hoher Aufwand für Betrieb und Wartung
- Budget reicht nicht für Ersatz oder Grundrenovierung von Alt-Systemen
- Druck zur Veränderung aus dem Geschäft und der IT-Technologie gleichermaßen
- Steigende Komplexität der notwendigen Veränderungen an IT-Systemen

Ziel der IT eines Unternehmens war und ist es, Geschäftsprozesse zu unterstützen und das Unternehmen wettbewerbsfähiger und wirtschaftlicher zu gestalten. Die unzureichende Anpassungsgeschwindigkeit der IT führten in den letzten Jahren zu Unzufriedenheit und so vermehrt zu einer Verlagerung der Arbeit (Outsourcing, SaaS, externe Services), ohne das Problem an der Wurzel zu lösen.

Das Kernproblem besteht in dem hohen Aufwand, der entsteht, wenn komplexe existierende IT-Systeme an veränderte Geschäftsprozesse anzupassen sind. Eine nachhaltige Problemlösung muss folglich an diesem Punkt ansetzen.

Bekannte Lösungen beschränken sich darauf, das Problem abzumildern oder zu verlagern:

- Die Auslagerung der Arbeit an Standorte mit billigen IT-Arbeitskräften
- Die Auslagerung von Teilen oder der gesamten IT an Spezialisten, die der Komplexität besser gewachsen sind
- Einsatz von Standardsoftware, womit die Gestaltung von Geschäftsprozessen umgangen wird

Zurzeit gleichen sich die Gehälter der IT-Spezialisten weltweit an; es ist daher zu erwarten, dass insbesondere die Auslagerung in Niedriglohnländer in Zukunft weniger lukrativ sein wird.

SOA, richtig eingesetzt, löst das Problem der unflexiblen IT grundlegender und nachhaltiger.

Service-orientierte Architekturen stellen zeitgemäße Konzepte zur Verfügung, um IT-Infrastrukturen schrittweise flexibel und anpassbar zu gestalten. Gleichzeitig werden Geschäftsprozesse und das Management der Fachbereiche direkter verbunden.

IT sollte dabei kein Selbstzweck und auch keine isolierte Insel im Unternehmen sein. Ihre Aufgabe ist es, mit möglichst kurzen Reaktionszeiten informationstechnische Strukturen aufzubauen und anzupassen, um alle zentralen Aspekte professioneller Betriebswirtschaft wie Performance-Management, Risiko-Management und Compliance-Anforderungen bestmöglich abzubilden und zu unterstützen. Wird aber speziell das Zusammenspiel von betriebswirtschaftlichem Management und Unternehmens-IT zum dominanten Faktor für

langfristigen Geschäftserfolg, stellen sich einige entscheidende Fragen:

- Wie löst man die scheinbare Unversöhnlichkeit von Betriebswirtschaft und IT, die bisher keine gemeinsame Sprache und Methoden entwickelt haben, dafür aber mit Vorurteilen und Misstrauen belastet sind?
- Wie schließt man den results gap – jene Kluft zwischen der erhofften Effizienz und Produktivität von Geschäftsanwendungen und dem tatsächlichen Nutzen für die Anwender im Unternehmen?
- Wie schließt man den IT gap – die Verzögerung, die aus der langsamen Reaktionszeit der IT auf betriebswirtschaftliche Anforderungen entsteht: Bis zum Lieferzeitpunkt haben sich die Anforderungen an IT-Services und -Applikationen bereits substanziell weiterentwickelt. Was als Lösung konzipiert wurde, ist schon bei der Implementierung unzureichend. Durch zum Teil veraltete Ansätze werden neue Probleme aufgeworfen, die letztlich Unzufriedenheit, mangelndes Vertrauen in die Leistungsfähigkeit der IT-Abteilung und langfristig hohe Kosten verursachen. Analysten wie Gartner schätzen, dass Unternehmen aktuell rund zwei Drittel der IT-Kosten in Maintenance (Wartung) statt in Neuinvestitionen und Zukunftssicherung investieren.
- Wie gelingt der Quantensprung der IT: Interoperabilität innerhalb der digitalen Welt und kurze Reaktionszeiten auf externe Einflüsse und Anforderungen? Dies betrifft insbesondere das betriebswirtschaftliche Management, die Berücksichtigung neuer gesetzgeberischer Herausforderungen wie Basel II oder Sarbanes-Oxley und andere wesentliche Bestimmungsfaktoren sich rasant entwickelnder Märkte.

Stehen diese Fragen im Zusammenhang mit SOA? Eine zukunftsweisende Geschäftsarchitektur muss sich diesen Fragen stellen und auch überzeugende Antworten geben!

Viele Geschäftsprozesse laufen heute IT-gestützt ab und sind in der Regel von mehreren in sich geschlossenen, transaktionsorientierten Anwendungen abhängig. Die Kommunikation zwischen den Anwendungen ist hinsichtlich der Datenformate und -typen häufig nicht standardisiert und zu fein granuliert. Traditionell oblag die Aufgabe der Integration der unterschiedlichen betriebswirtschaftlichen Silo-Anwendungen (ERP, CRM, SCM, etc.) innerhalb der Unternehmen den Enterprise-Application-Integration-Lösungen (EAI) und ihren diversen Adaptern und entsprechenden Entwicklern. Unzureichende Flexibilität und Agilität der Prozesse und damit der Unternehmen sowie enorme Wartungskosten sind das heutige Resultat.

Mit dem Aufkommen des Internets, des Service-orientierten Paradigmas und dessen zunehmenden Reifegrades entwickelte sich eine neue Art der Kommunikation und Integration für vernetzte systemübergreifende End-to-End-Geschäftsabläufe – auch über Unternehmensgrenzen hinweg (B2B). Die Hauptmerkmale dieser neuen Welt sind eine auf Standards beruhende Line-of-Business (LOB) oder auch semantische Interoperabilität zwischen den Anwendungen sowie deren lose Kopplung, die wichtigste Voraussetzung für flexiblere Software-Systeme.

Im Mittelpunkt der flexiblen Systeme stehen Services, die eine oder mehrere fachliche Funktionen beschreiben und durch Austausch von Business-Objekten in Form von strukturierten Nachrichten kommunizieren. Die fachlichen Services bilden dabei einen Software-Baustein mit eigenständigen fachlichen Funktionen. Die verwendete Programmiersprache (Java, C# oder C++) ist nebensächlich. Services haben eine Adresse und kommunizieren über standardisierte Nachrichten, die in der Regel mittels XML beschrieben werden. Die Schnittstellenbeschreibung von Services, das Schema, erfolgt beispielsweise mit der Webservices Definition Language (WSDL).

Fazit

- Der Markt verlangt immer kürzere Reaktionszeiten, um informationstechnische Strukturen aufzubauen und anzupassen.
- Bisherige Lösungen kommen aufgrund der stark ansteigenden Komplexität von verteilten IT-Systemen an ihre technische und ökonomische Leitungsgrenze.
- In großen Teilen der Branche besteht Einigkeit darüber, dass Lösungen basierend auf Service-orientierten Architekturen dieses Spannungsfeld entschärfen

und sich hinsichtlich der Kosten für Wartung und Weiterentwicklung von komplexen Systemen positiv auswirken.

2.3 Erfolgsfaktoren

Die Einführung einer unternehmensweiten SOA kann mit einer längeren Reise verglichen werden. Nach einem bekannten Sprichwort beginnt auch eine lange Reise mit dem ersten Schritt. Doch vor dem ersten Schritt sollte das Ziel bestimmt und die Richtung festgelegt sein. Zudem sollte jeder Reisende eine Vorstellung davon haben, was ihn auf dieser Reise erwartet, mit welchen Unwägbarkeiten er zu rechnen hat und wo besonders interessante und herausfordernde Wegstationen sind. Bewährt haben sich in diesem Zusammenhang Reiseberichte, die die Erfahrungen und Empfehlungen von anderen Reisenden zur Verfügung stellen.

Mit der SOA-Reise verhält es sich ähnlich. In der Vergangenheit wurden in zahlreichen Projekten viele Erfahrungen gesammelt und Erfolgsfaktoren haben sich herauskristallisiert. Dieses Kapitel befasst sich mit den wesentlichen Empfehlungen für die Vorbereitung und die ersten Schritte einer SOA-Transformation.

Erfolgsfaktor: Methoden des Änderungsmanagements anwenden

Die langfristige Zielsetzung von SOA führt zu grundlegenden Veränderungen im Unternehmen. Damit lassen sich die generellen Erkenntnisse und Regeln aus dem Änderungsmanagement anwenden. Insbesondere sind hierbei (in Anlehnung an John P. Kotters acht Schritte zum Veränderungserfolg) zu nennen:

- Ein Gefühl der Dringlichkeit erzeugen: Aufzeigen der Herausforderungen, denen mit SOA begegnet werden soll
- Eine Führungskoalition aufbauen: Ein starkes und sichtbares Management, das die Transformation treibt
- Vision und Strategien entwickeln: Definition eines idealen Endziels
- Die Vision des Wandels kommunizieren: Die SOA-Strategie und den Weg kommunizieren
- Empowerment auf breiter Basis: Aufmerksamkeit schaffen
- Kurzfristige Ziele ins Auge fassen/„short term wins“ generieren: Klein und überschaubar anfangen
- Erfolge konsolidieren und erreichte Verbesserungen systematisch weiter ausbauen: Präzises Projektmanagement und Aufbau einer „SOA Governance“
- Neue Ansätze in der Kultur verankern

Erfolgsfaktor: Ganzheitliche Betrachtung

Ein grundlegender Erfolgsfaktor für SOA ist eine ganzheitliche Betrachtung der Ausrichtung und Wertschöpfung des Unternehmens, denn SOA beinhaltet mehrere Aspekte:

- Unternehmens- und IT-Strategie
- Governance und Organisation
- Fachliche und technische Architektur

Erst das nahtlose Zusammenspiel aller Aspekte macht eine effiziente Realisierung einer SOA möglich, dabei bedarf es der Beteiligung des Managements. Jedem im Unternehmen sollte klar sein, was Serviceorientierung bedeutet, welche Facetten sie hat und welche Vorteile sich damit erzielen lassen. Dieses Wissen muss dann auf das eigene Unternehmen angewendet werden. Wo sind die Bereiche im eigenen Unternehmen, die besonders stark von einer SOA profitieren würden? Was soll dabei erreicht werden? Wo ist ein guter Startpunkt? Was wäre ein ideales Endziel? SOA ist kein Allheilmittel und nicht für jedes Problem oder für jeden Zweck geeignet. Die richtigen Anwendungsgebiete zu finden, ist eine der zentralen Aufgaben.

Erfolgsfaktor: Mit einem kleineren, überschaubaren Projekt anfangen, das sich an dem großen Gesamtbild orientiert und schnell umsetzbar ist.

Die Einführung kann nicht im Rahmen eines isolierten Großprojektes durchgeführt werden, das ein immenses Investment benötigt und große Teile der verfügbaren Ressourcen bindet. Die Transformation zu einem Service-orientierten Unternehmen hat Auswirkungen auf die Unternehmensstrategie, das Geschäftsmodell, die Geschäftsprozesse, die Unternehmenskultur, die IT-Anwendungen, die IT-Infrastruktur und die IT-Prozesse. All diese Segmente gleichzeitig anzugehen ist zu riskant, die Komplexität eines solchen Projekts ist kaum beherrschbar. Darüber hinaus muss vorrangig die Implementierung von neuen fachlichen Anforderungen gewährleistet sein. Oftmals haben diese kurzfristig eine höhere Priorität als die Umstrukturierung der Infrastruktur. Eine systematische und integrierte Vorgehensweise ist erforderlich. Für den schrittweisen Einstieg sind zwei Alternativen denkbar:

IT-Projekte, die neue Anwendungsfunktionalität bereitstellen, sollten gleichzeitig einen Teil der Architekturfunktionalität realisieren. Nur so verbinden sie Geschäftsnutzen mit Architekturnutzen im Sinne einer iterativen, evolutionären Migration.

Ein oder zwei Bereiche werden ausgewählt, mit denen begonnen wird. Dies könnte ein Geschäftsprozess sein, der durch Teilautomatisierung und höhere Änderungsflexibilität einen deutlichen Mehrwert für das Geschäft liefert und genügend Aufmerksamkeit im Unternehmen erreicht, um - dadurch angespornt - weitere Projekte auf den Weg zu bringen. Das Projekt sollte überschaubar und innerhalb weniger Monate umsetzbar sein. Da in dem ersten Schritt Erfahrungen gesammelt werden, sollte von dem Ergebnis nicht der Unternehmenserfolg abhängig sein.

Erfolgsfaktor: Einsatz von erfahrenen Mitarbeitern

Für die ersten Projekte ist der Einsatz von Mitarbeitern („Seniors“) ausschlaggebend, die bereits Erfahrungen bezüglich der Thematik und der Umsetzung solcher Projekte haben. Ist im Unternehmen kein ausreichendes Wissen vorhanden, sollte zusätzlich externe Expertise hinzugezogen werden, um den Erfolg zu gewährleisten. Das Projekt sollte sowohl Architekten, Entwickler als auch Vertreter der Fachabteilungen einschließen. Der Erfolg dieser Phase ist die beste Werbung und schafft Vertrauen für die nächsten Schritte.

Erfolgsfaktor: Bewerten des Nutzens nach jedem Schritt

Am Anfang gilt es möglichst viel und bewusst zu lernen und das gewonnene Wissen zu verbreiten. Auf Grundlage der Erfahrungen setzt man weitere Projekte auf und baut so Stück für Stück die Service-Landschaft auf. Dazu gehört auch eine Bewertung des erwarteten und erzielten Nutzens nach jedem Schritt, um zu erkennen, ob die gewählte Strategie richtig ist.

Erfolgsfaktor: SOA Governance

Der Nutzen einer SOA steigt erst mit dem Durchdringungsgrad des Unternehmens. Um vorhandene ineffiziente Ansätze zu verdrängen, müssen verbindliche Richtlinien und Prozesse festgelegt und konsequent durchgesetzt werden. Somit bekommt das prozessgesteuerte Architekturmanagement eine wesentliche Bedeutung für SOA. Darüber hinaus sind die fachliche Strukturierung, die Anpassung der Zuständigkeiten, sowie abgestimmte Prozesse zur Einbindung von Services notwendig, um eine SOA in komplexen Unternehmensstrukturen zu implementieren.

Sobald die SOA-Transformation anläuft, müssen die Organisationsstrukturen und die Prozesse zur Weiterentwicklung der SOA kontinuierlich überwacht und optimiert werden. Die Mechanismen, Rollen und Regeln dazu werden unter dem Schlagwort SOA Governance zusammengefasst. Unter Corporate Governance versteht man allgemein Steuerungs- und Regelungssysteme, die Unternehmen befähigen sollen, ihre Ziele zu erreichen. Diese Systeme sind von der Führungsorganisation und deren Prozessen zu unterscheiden und dienen ihrer Kontrolle. SOA Governance ist daher ein Teil der Corporate

Governance. Sie ist darüber hinaus Teil der IT Governance, die sicherstellt, dass die Konzepte und Prinzipien der Serviceorientierung auf angemessene Weise in ein Unternehmen eingeführt werden. Schließlich wacht sie darüber, dass die gesetzten Geschäftsziele für die definierten Services erreicht werden.

Um dies zu erreichen, müssen die Gremien und Rollen sowie deren zugrundeliegenden Richtlinien, Verfahren und Standards ebenso definiert werden, wie die Metriken und Hilfsmittel, die benötigt werden, um die SOA Governance Prozesse durchzuführen.

Erfolgsfaktor: Masterplan für den Aufbau der Infrastruktur

In dieser Phase der SOA-Umsetzung tritt die Infrastruktur in den Vordergrund. Anfangs ist der Einsatz von Enterprise Service Bus, Registry und Repository noch nicht zwingend. Mit steigender Servicezahl muss dafür rechtzeitig eine Lösung konzipiert werden. Dabei ist es wichtig auf zuverlässige Partner zu setzen, die auch über mögliche Hürden helfen. Die Infrastruktur muss dabei nicht aus einer Hand kommen. Wichtiger ist es, dass die eingesetzten Produkte auf die Unternehmenssituation abgestimmt sind und auf Standards aufsetzen.

Erfolgsfaktor: Verwendung von Standards und standardkonformer Produkte beim Aufbau der SOA-Plattform

Wesentlich für den Erfolg einer SOA, insbesondere über Unternehmensgrenzen hinweg, ist der Einsatz von Standards anstelle von proprietären Lösungen. Der Einsatz von Standards erleichtert zum einen die Interoperabilität und zum anderen können technologische Weiterentwicklungen mit standardkonformen Produkten ohne großen Mehraufwand genutzt werden. Häufig weichen Firmen von Standards ab, da sie befürchten, den Anforderungen des eigenen Unternehmens damit nicht begegnen zu können. Dabei besteht das Risiko, dass sich diese Unternehmen durch proprietäre Eigenentwicklungen von der technologischen Entwicklung des Marktes abkoppeln und dies mit hohen finanziellen Aufwänden kompensieren müssen.

Erfolgsfaktor: Kooperation zwischen Fachbereichen und IT

Das Konzept der Serviceorientierung setzt eine intensive Zusammenarbeit der Fachbereiche und des IT-Bereichs eines Unternehmens voraus. Die bislang häufig praktizierte Vorgehensweise der Fachbereiche, ihre Anforderungen zu definieren und diese anschließend der IT ohne entsprechende Kommunikation zur Umsetzung zu übergeben, funktioniert hier nicht.

Beispielsweise müssen die Geschäftsprozessmodelle von den Fachbereichen und der IT auf Basis ihrer unterschiedlichen Sichtweisen gemeinsam erstellt werden. Dies verhindert, dass zwei unterschiedliche Modelle entstehen. Bislang wurden die Prozessmodelle in den Fachbereichen häufig mit einem Grafikwerkzeug informell „gemalt“ und dann von IT-Experten für die maschinelle Ausführbarkeit formal modelliert. Diese unterschiedlichen Darstellungen bergen die Gefahr unterschiedlicher Interpretationen, die erst nach der Entwicklung der IT-Anwendungen sichtbar werden.

Service-orientierte Architekturen versprechen ein Ende der Disharmonien zwischen Business-Anforderungen und technischer Realisierung. Es entwickelt sich eine gemeinsame Begrifflichkeit. Damit die Potenziale von Service-orientierten Architekturen ausgeschöpft werden, muss im Rahmen des SOA-Veränderungsprozesses aktiv auf die enge Kooperation und Zusammenarbeit zwischen den Fachbereichen und dem IT-Bereich hingewirkt werden.

Werden die genannten Erfolgsfaktoren beachtet, können die großen bereits bekannten Stolperfallen vermieden werden. Trotz vieler Detailprobleme, die eine SOA-Transformation mit sich bringt, sollte durch sorgfältige Planung, gutes Projektmanagement und dem sukzessiven Erfahrungsgewinn aus jedem Projekt der Weg in ein Service-orientiertes Unternehmen von Erfolg gekrönt sein.

Fazit

Die Einführung einer Service-orientierten Architektur führt in den meisten Unternehmen zu weitreichenden Veränderungen, deshalb kann dieser Prozess mit den bekannten Strategien des Änderungsmanagements, angepasst an die speziellen Herausfordungen Service-orientierter Architekturen, erfolgreich bewältigt werden.

2.4 Potenzielle Risiken und geeignete Maßnahmen

Bei der Einführung einer SOA liegen die größten Hemmnisse nicht in der Technologie, sondern in den Köpfen der Menschen. SOA wird häufig als reine IT-Methodik aufgefasst und daher abgelehnt. Bei SOA handelt es sich jedoch wie bereits erläutert um eine neue Ausrichtung der Unternehmensorganisation, die ihre Vorteile in der effektiveren Abstimmung der beteiligten Systeme wie Mensch, Unternehmenssteuerung und IT hat.

Die erfolgreiche Einführung moderner Software-Services hängt vor allem davon ab, wie gut Kommunikationslücken zwischen der Prozessebene und dem Applikationsdesign geschlossen werden können. Es empfiehlt sich bereits von Anfang an, alle Bereiche einzubinden.

Organisatorische und zwischenmenschliche Einflüsse

Keine kurzfristigen Kosteneinsparungen

Die ersten SOA-Projekte innerhalb eines Unternehmens führen in der Regel noch nicht zu Kosteneinsparungen. Gegebenenfalls erfordern sie sogar erhöhte Investitionen. Kosteneinsparungen und verkürzte Innovationszyklen sind erst spürbar, wenn Services wiederverwendet werden können bzw. wenn ihre Orchestrierung bereits besteht und nur angepasst werden muss.

Risiken

Die ersten SOA-Projekte eines Unternehmens können scheitern, weil die Unterstützung durch das Management und die Fachabteilungen während der Projekte nachlässt. Durch SOA werden keine kurzfristigen Kosteneinsparungen erreicht.

Geänderte Verantwortlichkeiten

Die Einführung einer SOA führt auch dazu, dass Verantwortlichkeiten neu strukturiert und angepasst werden. Die alte Ausrichtung an Applikationen und Fachbereichen muss gegebenenfalls aufgebrochen werden. Eine Kultur offener und konstruktiver Kommunikation ist erforderlich.

Risiken

- Die Umstrukturierung der Verantwortlichkeit kann dazu führen, dass Mitarbeiter die Projekte blockieren. Die Wahl von Services kann durch Kompetenzstreitigkeiten behindert werden, was den Nutzen mindert.
- Outsourcing und Automatisierung können den Verlust von Zuständigkeiten und Machtpositionen nach sich ziehen, eventuell kostet dies auch Arbeitsplätze.

Know-how bezüglich Legacy-Applikationen

Wenn Legacy-Applikationen – also ältere produktive Anwendungen – in eine SOA eingebunden werden sollen, erfordert die Planung einen genauen Überblick über die fachliche und die technische Architektur der Applikationen. Dies ist Voraussetzung, um fachlichen Nutzen und technischen Aufwand bei der Umsetzung von Services abzuwägen.

Risiken

- Das erforderliche Know-how ist nicht in den Unternehmen vorhanden bzw. Mitarbeiter mit entsprechendem Know-how halten dieses zurück, um ihren Aufgabenbereich zu schützen. Unzureichende Analysen führen zu nutzlosen Services bzw. zu dem technischen Scheitern der Projekte.
- Undurchsichtige heterogene IT-Landschaft
  Neben der zentralen Unternehmenssoftware wird häufig zusätzliche nicht integrierte Spezialsoftware eingesetzt. Lieferanten und Geschäftspartner setzen ebenfalls ihre eigene Unternehmenssoftware ein und möchten nicht auf eine zentrale Software migrieren. Darüber hinaus sind die unterschiedlichen Systeme oft hinsichtlich der Datenformate inkompatibel. So fehlt häufig eine nutzbare Datenbasis des Potenzial neuer Geschäftsfelder bzw. das Kaufverhalten der Kunden

Sachkenntnis und Erfahrung von Mitarbeitern

Sowohl auf dem Gebiet der Geschäftsprozesse als auch in der IT ist es häufig notwendig, externes Know-how "einzukaufen" und dieses dann an die Mitarbeiter weiterzugeben.

Risiken

Neben dem Risiko der Abhängigkeit von Beratern ist das Akzeptanzrisiko bei den eigenen Mitarbeitern zu beachten.

Auf dem Gebiet der IT muss ein Wechsel der Denkmuster erfolgen: weg von silogeprägten funktions- und anwendungsorientierten Systemen hin zu mehr service- und prozessorientierten Vorgehensweisen. Die Ausweitung der Softwareentwicklungsplattformen sowie die Integration der Geschäftsprozessmodellierung und der Code-Generierung haben Auswirkungen auf zukünftige Arbeitsteilungen im Softwareentwicklungsprozess. Solution-Architekten, welche die betriebswirtschaftlichen Hintergründe eines Prozesses verstehen, gestalten in verstärktem Maße die den gesamten Softwareentwicklungszyklus – vom Anforderungsmanagement über die Code-Generierung bis hin zum Betrieb.

Risiken

Denkschemata, die sich an Modellierungstechniken aus der Softwareentwicklung orientieren (z. B. UML), beziehen die Fachbereiche nicht ausreichend in die Planung ein.
Divergierende Begriffswelten, uneinheitliche Dokumentation der Services und mangelhafter Überblick verhindern oder erschweren die Verwendung bestehender Services und führen zu Redundanzen und verlängerter Time-to-Market.

Risiken des SOA-Architektur-Paradigmas

SOA ist keine Technologie, sondern ein Architekturparadigma, das der verstärkten Anforderung nach evolutionsfähigen Software-Systemen geschuldet ist. Um die Chancen und Risiken einer SOA beurteilen zu können, müssen das Architekturparadigma an sich und die Architekturentscheidungen, die bei der konkreten Realisierung zu treffen sind, getrennt betrachtet werden.

Risiken

- Sicherheit
  Bei einer großen Zahl der an übergreifenden Geschäftsprozessen Beteiligten herrschen Vorbehalte, ihre Daten könnten in falsche Hände geraten. Bei der Einbeziehung externer Dienste und der Bereitstellung von Diensten für externe Partner müssen Sicherheitskonzepte frühzeitig in die Entwicklung integriert werden. Sicherheit erfordert sowohl finanzielle als auch technische Ressourcen.
- Skalierbarkeit
  Die Bestandteile einer SOA, die die eigentliche Funktionalität erbringen, können auf unterschiedlichen Hardware-Plattformen betrieben werden. Eine SOA ist deshalb – per definitionem skalierbar. Es ist jedoch eine sorgfältige Analyse des Service-Providers erforderlich, um nicht andere nicht funktionale Anforderungen bei der Skalierung negativ zu beeinflussen.

- Test
  Das Testen eines Systems, das auf Basis einer SOA realisiert wurde, ist ein komplexes Problem. Teststrategien für herkömmliche Systeme können nicht unverändert übernommen werden, es entsteht Anpassungsbedarf. Um eine SOA erfolgreich zu testen, müssen die folgenden Fragen geklärt werden:
  - Wie sind die Systembestandteile im Netz verteilt?
  - Wie und welche Dienste können zur Laufzeit gebunden werden?
  - Welche konkrete Instanz eines Dienstes wird verwendet?
  - Wie erhält der Nutzer eines Dienstes die Informationen, die einen ausführlichen Test der Dienstnutzung in der SOA erlauben? Hat die Organisation ggf. Zugriff auf den Quellcode des Service, sodass daraus die Testfälle abgeleitet werden können?
  - Welchen Informationen über die Dienstnutzung müssen in Log-Dateien gesammelt werden und wo werden diese Dateien abgelegt?
  - Kann der Dienstbetreiber Logging-Informationen für die Dienstnutzer bereitstellen?
  - Stellt der Dienstbetreiber eine Test-Instanz des Dienstes bereit, die auf einem genügend komplexen Testdatenbestand basiert, sodass nicht auf Produktionsdaten zugegriffen werden muss?

  Das Risiko eines SOA-Tests besteht darin, dass Dienste nur unvollständig getestet werden (können) und so Fehler im Produktionsbetrieb nur sehr schwer zu lokalisieren sind.
- Performance
  Der Aufruf eines Service in einer SOA kann im Vergleich zu klassischen Anwendungen zusätzliche Ressourcen erfordern, da die Services im Netz verteilt sein können. Weitere Ressourcen werden für die Interpretation von orchestrierten Diensten und je nach Realisierung der Plattform Aufwand für das Routing von Dienstaufrufen benötigt. Wird das bei der Realisierung der Architektur nicht berücksichtigt, können SOA-basierte Anwendungen schnell inperformant werden.

Risiken konkreter Realisierungsentscheidungen

Bei der Realisierung eines SOA-basierten Systems sind verschiedene Architekturentscheidungen zu treffen, die jeweils ihre spezifischen Risiken bergen.

Nachrichtenbasierte vs. prozedurale Schnittstellen

Ein Prozeduraufruf erfordert die genaue Kenntnis der Reihenfolge und der Typen der einzelnen Parameter. Im Gegensatz dazu sind nachrichtenbasierte Schnittstellen sehr einfach gestaltet. Die zu bearbeitende Information ist in den Nachrichten kodiert und der Konsument einer Nachricht kann dynamisch entscheiden, wie diese zu interpretieren ist. Nachrichten sind viel weniger stark typisiert als Aufrufe von Prozeduren. Die Kopplung zwischen einzelnen Elementen einer Architektur kann so reduziert werden. Der Einsatz nachrichtenbasierter Schnittstellen führt somit zu einer Entkopplung auf der Ebene der Anwendungen.

Risiken

Der Verzicht auf strikte Typisierung kann leicht zu Fehlern führen, die erst zur Laufzeit identifiziert werden können. Auch wenn die Schnittstelle für Dienste nachrichtenbasiert ist, sollte bei der Realisierung von Diensten nicht auf die Vorteile der strikten Typisierung verzichtet werden.

Granularität von Schnittstellen

Eng verbunden mit der Auswahl des Typs einer Schnittstelle ist deren Granularität. Je feinkörniger die Funktionalität in einer Schnittstelle spezifiziert ist, desto stärker sind der Nutzer und der Anbieter der Funktionalität gekoppelt und desto größer ist der Aufwand bei der Kommunikation mit im Netzwerk verteilten Elementen einer Architektur.

Bei der Realisierung von Diensten durch Aggregation oder Komposition, d. h., wenn von einer lokalen Kommunikation auf einer Plattform ausgegangen werden kann, ist die Verwendung feinkörniger prozeduraler Schnittstellen angebracht. Sobald jedoch Funktionalität lokationstransparent angeboten werden soll, eine entfernte Kommunikation also nicht ausgeschlossen werden kann, sollten grobkörnige Schnittstellen und gegebenenfalls auch eine nachrichtenbasierte Kommunikation realisiert werden [ACM2003].

Risiken

Feine Granularität der Services führt zu viel Last auf dem Enterprise Service Bus und kann in Performance-Problemen resultieren.
Feingranulare Services können sehr flexibel wiederverwendet werden. Sie realisieren jedoch wenig Fach-Logik und stellen somit einen geringeren Investitionswert dar als komplexere Services. Ihre Verwaltung und Dokumentation stellt eine neue Herausforderung dar.
Zu grobe Granularität führt dazu, dass Services zu große Anteile eines Geschäftsprozesses abdecken und somit für die Komposition in anderen Geschäftsprozessen nicht mehr verwendet werden können.

Synchrone vs. asynchrone Kommunikation

Die Verwendung asynchroner Kommunikationsmechanismen führt zu einer Entkopplung auf der Prozess-Ebene.

Ein Client, der eine sofortige Rückantwort auf seine Dienstanforderung – möglichst über denselben Kommunikationskanal – fordert, ist viel stärker an den Dienst-Anbieter gebunden als einer, der asynchron kommuniziert und im Rahmen seiner Dienstanforderung den Kanal für die Rückantwort spezifiziert.

Werden verlässliche Kommunikationskanäle wie z. B. Messaging Services verwendet, müssen Client und Dienstanbieter nicht zur selben Zeit aktiv sein, d. h., der Ausfall eines Dienstanbieters führt nicht automatisch zum Abbruch des zu unterstützenden Prozesses. Werden asynchrone Kommunikationsmechanismen mit nachrichtenbasierter Kommunikation verbunden, so sind die beteiligten Anwendungen sowohl auf Anwendungs- als auch auf Prozess-Ebene entkoppelt.

Wenn es die Logik des zu unterstützenden Verwaltungsprozesses erlaubt, sollte die Kommunikation zwischen verteilten Architekturbestandteilen asynchron erfolgen, innerhalb der Realisierung ist eine synchrone Kommunikation vorzuziehen.

Risiken

Asynchrone Kommunikation erfordert einen speziellen Programmierstil, da nicht mehr davon ausgegangen werden kann, dass Service-Nutzer und Service-Erbringer zur selben Zeit aktiv sind, dass die Service-Anforderung sofort ausgeführt und das Ergebnis sofort zurückgeliefert wird. Die Komplexität dieser Anwendungen erhöht sich und stellt neue Herausforderungen an die Entwickler.

Implementierungs- und Verteilungsmodell für Architekturbestandteile

Um zu gewährleisten, dass die Realisierungsgarten von Architekturbestandteilen bei unveränderten Schnittstellen ausgetauscht werden können, muss ein Implementierungsmodell für die Architekturbestandteile spezifiziert werden. Das Implementierungsmodell muss konform zur gewählten Referenzarchitektur sein und die folgenden Vorgaben enthalten:

Art der Spezifikation der Schnittstelle(n)

Die Spezifikation einer Schnittstelle enthält neben der Funktionalität in Abhängigkeit von der gewählten Referenzarchitektur noch weitere Informationen. Bei einer komponentenbasierten oder objektorientierten Architektur muss z. B. die Reihenfolge beschrieben werden, in der die einzelnen in der Schnittstelle beschriebenen Funktionen aufgerufen werden können. Daneben müssen Verhaltensaspekte der einzelnen Schnittstellenfunktionen dokumentiert werden.
Im Kontext einer diensteorientierten Architektur auf der Basis einer Web-Service-Plattform muss die Funktionalität der Schnittstelle mittels WSDL beschrieben werden. Für die anderen Referenzarchitekturen sind beliebige Schnittstellenbeschreibungssprachen zulässig. Zur Beschreibung des Aufrufprotokolls und zur Beschreibung von Verhaltensaspekten existieren noch keine Vorgaben.

- Vorgaben zur Implementierung der spezifizierten Funktionalität
- Format der Auslieferung eines Architektur-Elements
- Vorgaben zur Installation eines Architektur-Elements

Als ein Beispiel sei auf die Vorgaben der Service Component Architecture für Java-zentrierte Realisierung [SCA2005] bzw auf [MSDN Solution Architecture Center] für eine .Net-konforme verwiesen.

Risiken

Wird das Implementierungs- und Verteilungsmodell nicht spezifiziert, so können die Implementierungen der Dienste nicht ausgetauscht werden.

Weiterführende Links

MSDN Solution Architecture Center
http://msdn.microsoft.com/architecture

The Architecture Journal
http://www.architecturejournal.net/

Microsoft Patterns & Practices
http://msdn.microsoft.com/practices/

Literatur

[ACM2003] D. Alur, J. Crupi, D. Malks, »Core J2EE-Pattern – Best Practices and Design Strategies«, 2nd Edition, Prentice Hall, 2003

[SCA2005] Joint Whitepaper by BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase,» Service Component Architecture«, Version 0.9, 2005

[BW2006] Whitepaper der Evidon GmbH »SOA – ein pragmatischer Ansatz« Dr. M. Bark, W. Wulf, Vers. 1.3 2006

3 Vorbereitung und Planung

3.1 SOA Readiness

Unternehmen müssen sich grundsätzlich die Frage stellen, wie Service-orientierte Architekturen eingeführt werden können. Wird die SOA-Technologie beispielsweise im Rahmen eines Projektes eingeführt, liefern die Projektziele die entsprechenden Anforderungen. In anderen Kontexten ist die Durchführung einer Prüfung und Bewertung des SOA Reifegrades des Unternehmens, der sogenannten SOA-Readiness, sinnvoll: So etwa, wenn Unternehmen SOA auf Unternehmensebene zur strategischen Ausrichtung einführen wollen oder grundsätzlich die Potenziale einer SOA-Einführung ermitteln wollen.

SOA Readiness ermittelt und bewertet die Voraussetzungen für die Einführung oder Weiterentwicklung einer SOA in einem Unternehmen. Darüber hinaus werden die Chancen und Herausforderungen bei der Einführung einer SOA identifiziert.

Zur Ermittlung der SOA Readiness werden üblicherweise Assessments durchgeführt. Darüber hinaus kann ein SOA-Maturity-Modell, s. u., zur Klassifizierung der SOA Readiness verwendet werden.

Die Einführung einer SOA kann Auswirkungen auf alle Unternehmensbereiche haben, daher sollten zur Ermittlung der SOA Readiness alle potenziell involvierten Bereiche betrachtet werden. Ein Assessment kann aber auch in eingeschränkter Form durchgeführt werden und inkrementell bei wachsenden Anforderungen erweitert werden.

Folgende Faktoren innerhalb eines Unternehmens werden analysiert und bewertet:

- Geschäftsmodelle und -prozesse des Unternehmens
- Leistungen und Services der IT
- IT-Technologie wie Anwendungen und Infrastruktur
- Prozessorientierung
- Organisation
- Betrieb

SOA-Assessements werden in der Regel auf Basis von Fragebögen und Workshops mit den entsprechenden Interessenvertretern durchgeführt.

Indikatoren für SOA Readiness

SOA Awareness & Skills

Grundlage für das erfolgreiche Umsetzen von SOA-Projekten ist neben den benötigten Skills eine offene Unternehmenskultur, die über die notwendigen Veränderungen sowie die neuen Strukturen, die sich durch eine SOA ergeben, regelmäßig berichtet. Wichtig ist dabei, dass die Veränderungen nicht nur theoretisch vorgestellt werden, sondern fester Bestandteil der Unternehmenskultur werden.

Die folgenden Kriterien müssen für die erfolgreiche Umsetzung von SOA-Projekten und -Prozessen erfüllt werden:

Management Support

- Hohe Awareness innerhalb der Geschäftsleitung (Vorstand)
- Klares Management-Committment zur Einführung von Geschäftsprozessen auf der Basis von SOA
- Ein Management-Mitglied sollte Mentor der SOA-Projekte sein
- Das Management sollte im Steering Committee der SOA-Projekte verankert sein
- Klare SOA-Vision durch das Management

Transparenz & Kommunikation

- Die langfristigen SOA-Visionen sowie die SOA-Projekte sollten klarer Bestandteil der Unternehmensstrategie sein, eine SOA-Roadmap entwickelt und regelmäßig kommuniziert werden
- Die SOA-Projekte sollten regelmäßig im Intranet dargestellt werden
- Klare Verantwortlichkeiten und Ansprechpartner müssen zugewiesen werden

- Die SOA-Projekte sind fester Bestandteil der Unternehmenskommunikation

Change Management

- Die Mitarbeiter müssen frühzeitig und regelmäßig über die mit den SOA-Projekten einhergehenden Veränderungen informiert werden.
- Die Vorteile und Notwendigkeit eine SOA einzuführen, müssen dargestellt werden.
- Die Mitarbeiter müssen frühzeitig auf die bevorstehenden Veränderungen vorbereitet werden, entsprechende Schulungs- und Unterstützungsmaßnahmen müssen initiiert werden.

Prozessverständnis

- ist die Grundlage, um zu identifizieren, welche Prozesse durch eine SOA optimiert werden können.
- ist die Grundlage, um die Prozesse erfolgreich auf eine SOA abbilden zu können sowie die potentiellen Vorteile durch eine SOA auch wirklich zu heben.
- ist die Grundlage, um klare Anforderungen an die technische Umsetzung der SOA definieren zu können.

Technologieverständnis

- Ein hohes Maß an Technologieverständnis ist notwendig, um die besten Konzepte und Produkte entsprechend den individuellen Anforderungen zur Umsetzung einer SOA zu berücksichtigen.

Nutzenverständnis

- Um den Vorteil einer SOA überzeugend darstellen zu können, muss eine Nutzenbetrachtung mit den klaren Vorteilen, die mit der SOA einhergehen, erarbeitet werden.
- Ein Business Case ist notwendig, um die Optimierungspotenziale zu quantifizieren.
- Die kurz-, mittel- und langfristigen Vorteile durch SOA müssen dargestellt werden und entsprechend den jeweiligen Unternehmensherausforderungen betont werden.

Organisation

Die Einführung, der Betrieb sowie die kontinuierliche Weiterentwicklung einer Service-orientierten Architektur gehen nach den bisherigen Erfahrungen und wie in den vorangegangenen Abschnitten bereits erwähnt weit über technische Maßnahmen hinaus. Unter dem Begriff SOA Governance werden die vielfältigen Querbeziehungen der unterschiedlichen Bereiche zusammengefasst. Ein wesentlicher Aspekt ist die organisatorische Abstimmung zwischen IT-Abteilung und den an den Geschäftsprozessen beteiligten Fachbereichen. So, wie auf technischer Ebene Anwendungs- und IT-Silos aufgebrochen werden, damit diese flexibel interagieren können, müssen auch organisatorische Hemmnisse überwunden werden, um eine nachhaltige Flexibilität des Unternehmens auf einer betriebswirtschaftlich sinnvollen Basis zu erzielen.

SOA – Zusammenspiel zwischen Fachabteilungen und IT

Eine erfolgreiche SOA-Umsetzung dient insbesondere der Verbesserung der Kommunikation. Dies steigert darüber hinaus die Umsetzungsgeschwindigkeit zwischen den Geschäftsbereichen und den IT-Bereichen, die die Anforderungen des Managements bzw. der Fachabteilungen umsetzen – d. h. organisatorisch sind hier entsprechende Schnittstellenkompetenzen zu implementieren.

Ziel ist es, die Kompetenzen für Business Services so weit wie möglich zu dezentralisieren und insbesondere die Fachbereiche frühzeitig einzubeziehen. Gleichzeitig sollten die zur Umsetzung benötigten Infrastruktur-Services entsprechend zentralisiert werden, um notwendige Skaleneffekte, etwa durch Virtualisierungstechniken, zu erzielen. Ein solches Vorgehen bietet die Chance, Infrastruktur-Kompetenz kostengünstig zu etablieren und dennoch einen passgenauen Zuschnitt der Business Services auf die unterstützten Geschäftsprozesse zu ermöglichen.

Insbesondere bezüglich der zentralisierten Infrastruktur-Services stellt sich die Frage der Finanzierung. In diesem Kontext ist es wichtig, mit den Infrastrukturkosten

bewusst umzugehen und den Business Case nicht „schön zu rechnen". Vor der Umsetzung muss auf Management-Ebene abgestimmt werden, ob die Kosten für die Infrastruktur als zentrale IT-Kosten gebucht werden, ein mehr oder weniger komplexes Umlageverfahren (Cross-Unit Funding) angewendet wird oder beispielsweise ein Pilotprojekt einen entscheidenden Teil der Kosten tragen kann. Eine wichtige organisatorische Voraussetzung für die SOA Readiness ist ein für alle Beteiligten transparentes Verfahren.

Organisatorische Aspekte für eine erfolgreiche SOA-Implementierung

Zwei grundlegende organisatorische Voraussetzungen für eine erfolgreiche SOA-Implementierung sind a) die Möglichkeit, Architekturfragen zentral anzugehen und b) das Vorhandensein entsprechenden SOA-Know-hows in den Fachabteilungen – also dezentral. Hier einige ausgewählte Best-Practice-Beispiele:

- Zusammenarbeit mit einem Enterprise-Architekten bzw. die Etablierung eines Enterprise-Architecture-Kompetenzzentrums mit der Aufgabe, eine unternehmensweit einheitliche Sicht auf die verschiedenen SOA-Projekte zu gewährleisten
- Aufbau von SOA-Architektur-Know-how in den Fachbereichen: Es ist notwendig, Business-Analysten auszubilden, die als kompetente fachliche Ansprechpartner sowohl mit den Enterprise-Architekten als auch später mit den entsprechenden Solution- bzw. Software-Architekten zusammenarbeiten können.
- Aufbau von Infrastruktur-Kompetenz(-zentren): Die durch die verbesserte Kommunikation zwischen Fach- und IT-Abteilung gewonnene Agilität muss durch eine entsprechend flexible und ressourcenoptimierende Infrastruktur unterstützt werden.

Weitere Aspekte betreffen beispielsweise die Gestaltung der Entwicklungs- und Betriebsprozesse:

- Definition von Richtlinien zur Identifikation von Services, die konform mit der etablierten SOA-Infrastruktur sind
- Festlegung von abteilungsübergreifenden Change- und Koordinationsprozessen mit entsprechenden Verantwortlichen
- Projektübergreifend einheitliche Verwaltung von Metadaten sowie Festlegung, wie mit diesen umzugehen ist
- Standardisiertes Schnittstellendesign und Contract Management

IT-Systemlandschaft und -Infrastruktur

Service-orientierte Architekturen sollten die bestehende IT-Infrastruktur soweit wie möglich nutzen. Darüber hinaus sind üblicherweise weitere SOA-spezifische Infrastrukturkomponenten vorzusehen, die den besonderen Bedürfnissen der SOA-Entwicklung und des SOA-Managements Rechnung tragen. Schließlich ist zu überlegen, wie IT-Komponenten, die sich a priori nicht in die SOA integrieren lassen, über Adapter- oder Wrapping-Konzepte dennoch für die SOA genutzt werden können.

Eine SOA lässt gemäß folgendem allgemeinen Komponentenmodells in Ebene oder Schichten strukturieren:

Ebene 1: Basis-Infrastruktur

Hier finden sich elementare IT-Komponenten wie etwa Server, Netzwerk, Speichersysteme. Um auf dieser Ebene die notwendige Agilität zu gewährleisten, kommen zunehmend Virtualisierungskonzepte zum Einsatz, die eine flexible Zuordnung von Ressourcen zu Services ermöglichen.

Ebene 2: Enterprise Components

Auf dieser Ebene werden IT-Systeme als elementare Service-Lieferanten verstanden.

Ebene 3: Integration, Enterprise Service Bus

Die Integration der Enterprise Components mittels geeigneter Middleware ermöglicht die Definition der abstrakteren Enterprise Services auf der nächsten Ebene. Diese Schicht wird häufig auch als Enterprise Service Bus bezeichnet. Werkzeuge aus dem Enterprise-Application-

Integration- (EAI) und Message-Queuing-Umfeld kommen hier ebenfalls zum Einsatz.

Ebene 4: Enterprise Services

Die Ebene der Enterprise Services beinhaltet einen Pool aus wiederverwendbaren Services, die zentral verwaltet werden.

Ebene 5: Orchestration, Business Process Management

Hier werden die Enterprise Services so zusammengeschaltet, dass sie Geschäftsprozesse geeignet abbilden. Ein wichtiges Qualitätsmerkmal einer SOA ist es, die abgebildeten Geschäftsprozesse entsprechend agil auf neue Anforderungen anpassen zu können.

Ebene 6: Präsentation

Schließlich ist auch die Interaktion mit dem Benutzer durch geeignete Services zu realisieren. Beispielsweise muss es möglich sein, für Multi-Channel-Architekturen nicht nur graphische Benutzeroberflächen, sondern auch browserbasierte Benutzerschnittstellen und Darstellungen für mobile Endgeräte umzusetzen.

Darüber hinaus sind orthogonal zu diesen Ebenen Querschnittsfunktionen für die Entwicklung (Design-Time) und das Management (Run-Time) vorzusehen. Hier sind vor allem Repository- und Registry-Services zu nennen. Schließlich ist das System-Management um ein entsprechendes Service-Management zu erweitern, das unter anderem Service-Level-Agreements überwacht und provisioniert.

Ein Ziel bei der Umsetzung von SOA ist die homogene Sicht auf eine üblicherweise heterogene IT-Systemlandschaft. Hilfreich für die Ermittlung der SOA Readiness ist deshalb auch der Einsatz eines einheitlichen Informationsmodells – eventuell unter Nutzung semantischer Integrationsstandards. Letztlich führen SOA-Initiativen immer zu Fragen, die sowohl die IT als auch die SOA Governance betreffen.

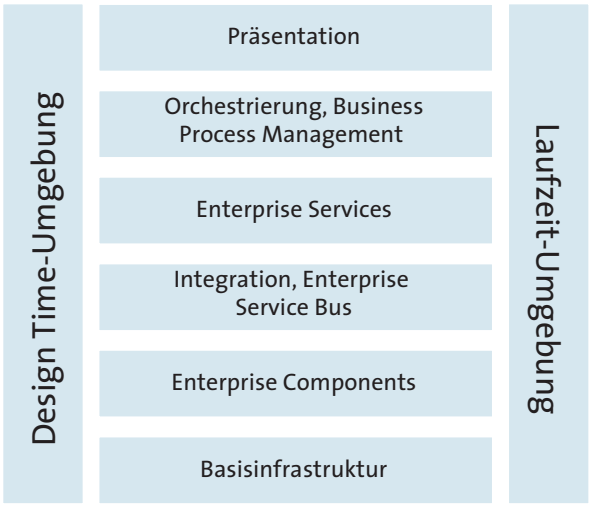


Abbildung 1: SOA-Infrastruktur

SOA-Maturity-Modell:

Folgende Faktoren sind also maßgeblich für die Entscheidung ob und wie eine SOA eingeführt bzw. erweitert werden soll: Stand der Kompetenz, die aktuelle Infrastruktur sowie das Bekenntnis des Managements zu einer SOA. Die Klärung dieser Fragen führt zu einer ersten Einschätzung, welchen Status Service-orientierte Architekturen zu einem bestimmten Zeitpunkt in einen Unternehmen einnehmen und was die Zielsetzung für zukünftige SOA-Initiativen sein kann.

Diese Einschätzung muss im Rahmen einer Gap-Analyse konkretisiert werden. Die Gap-Analyse kann im Rahmen eines SOA-Assessments erfolgen. Das Ergebnis des SOA-Assessments spiegelt die Diskrepanz zwischen dem aktuellen SOA-Status in einen Unternehmen sowie der Zielsetzung zur Einführung und Umsetzung weiterer SOA-Aktivitäten wider. Eine realistische Einschätzung zur Einführung von SOA in einem Unternehmen bildet die Basis für eine detaillierte Projektplanung zur Umsetzung der nächsten SOA-Schritte. Diese wiederum ist die Voraussetzung für die Etablierung einer erfolgreichen SOA-Roadmap.

Grundlage für die SOA-Gap-Analyse bildet das SOA-Maturity-Modell. Das SOA-Maturity-Modell ermöglicht eine erste Bewertung anhand der folgenden Kriterien:

- SOA-Kompetenzen
- SOA-Infrastruktur (Prozesse, Tools)
- Erfahrungen und erfolgreiche Projekte im Umfeld von SOA
- Strategische Verankerung innerhalb des Unternehmens (Competence Center, ...)
- Wie wichtig ist SOA als Konzept zur Umsetzung der Geschäftsstrategie, ist es ein elementarer Bestandteil?
- SOA-Geschäftsprozess-Architektur
- SOA-IT-Referenzarchitektur
- Aktuelle SOA-Initiativen
- SOA-Roadmap
- Kosten-/Nutzenanalyse auf Basis von SOA
- SOA Governance
- SOA-Leitprinzipien
- Change Management

Je nach den unternehmensspezifischen Erfordernissen und der Bedeutung für das Unternehmen erstreckt sich ein entsprechendes Assessment über einen Zeitraum von einem Tag bis zu mehreren Wochen. Drei Assessmentstufen werden empfohlen:

- 1 – 2-tägiger Workshop: Die wichtigsten Erfahrungen und Fähigkeiten des Unternehmens im Umfeld von SOA werden aufgenommen, eine erste Einschätzung des Unternehmens wird vorgenommen. Erste Ziele und Möglichkeiten auf Basis einer SOA werden aufgezeigt.
- 1 – 2-wöchiges Beratungsprojekt: Erfahrungen, Fähigkeiten, Prozesse und Projekte im Umfeld von SOA werden eruiert. Auf Basis dieser umfangreichen Analyse werden die Stärken und Schwächen des Unternehmens in diesem Kontext aufgezeigt. Konkrete SOA-Projekte zur Einführung einer Service-orientierten Infrastruktur, konkrete Geschäftsprozesse auf Basis von SOA sowie eine SOA-Roadmap werden vorgeschlagen.
- 4 – 8-wöchiges Assessment-Projekt: Sämtliche Erfahrungen, Fähigkeiten, Prozesse, Initiativen und Projekte im Umfeld von SOA werden wie im Beratungsprojekt umfassend eruiert. Auf Basis dieses sehr umfangreichen Assessments werden die Stärken und Schwächen des Unternehmens in diesem Umfeld detailliert aufgezeigt. Konkrete SOA-Projekte zur Einführung einer Service-orientierten Infrastruktur und zur Abbildung ausgewählter Geschäftsprozesse auf Basis von SOA sowie eine SOA-Roadmap werden vorgeschlagen und anhand einer Prioritätenliste bewertet. Konkrete Sofortmaßnahmen werden mit den Zuständigen im Unternehmen besprochen und sind Bestandteil einer langfristigen SOA-Unternehmensstrategie.

Eine *SOA-Roadmap* beschreibt die konkreten Schritte zur Umsetzung der SOA-Vision. Sie enthält die wesentlichen Aktivitäten und Zeitvorgaben:

Die Ergebnisse des SOA-Assessments werden in einem sechsstufigen SOA-Maturity-Modell abgebildet:

- Level 0 – Legacy Applications
  - Das Unternehmen hat bislang keine SOA-Aktivitäten begonnen oder Erfahrungen in diesem Umfeld gesammelt.
- Level 1 – SOA Strategy Initiation
  - Erste SOA-Initiativen wurden gestartet.
  - Überlegungen zur Verwendung von SOA zur Optimierung von Geschäftsprozessen oder der IT-Infrastruktur sind im Gange.
  - Erste SOA-Kompetenzen sind vorhanden.
- Level 2 – SOA Governance
  - Im Unternehmen sind bereits weitgehende SOA-Initiativen gestartet worden.
  - Erste Verantwortlichkeiten für die Umsetzung weiterer SOA-Strategien sowie zur Verankerung der zukünftigen SOA-Projekte (Governance) wurden festgelegt.
  - SOA-Projekte zur Abbildung von Geschäftsprozessen sind etabliert.
- Level 3 – Process Modeling
  - Erste SOA-Projekte befinden sich vor dem Abschluss.
  - Die strategischen Planungen zu SOA gehen in konkrete Geschäftsprozessimplementierungen über.
  - SOA ist fester Bestandteil der Unternehmensstrategie.
  - Erste wiederverwendbare Services wurden implementiert und im Rahmen der Projekte orchestriert

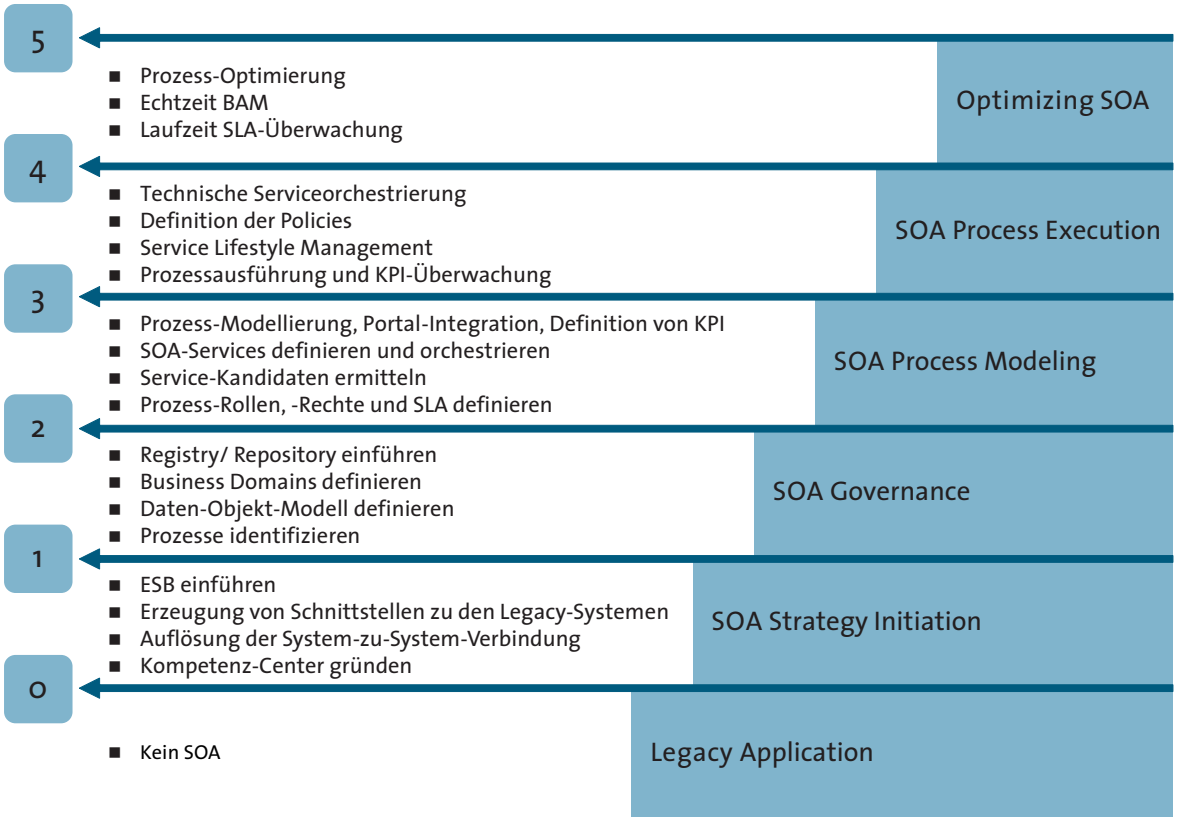


Abbildung 3: SOA-Roadmap/Maturity-Modell

- Level 4 – SOA Process Execution
  - Erste SOA-Aktivitäten wurden bereits erfolgreich durchgeführt.
  - Erste Geschäftsprozesse basieren auf SOA.
  - SOA ist Grundlage für die zukünftige Entwicklung neuer Geschäftsprozesse sowie der Optimierung bestehender Geschäftsprozesse.
  - Service Lifecycle Management ist als integrale Aufgabe zur Pflege der Services etabliert.

- Level 5 – Optimizing SOA
  - SOA ist ein integraler Bestandteilteil der Unternehmensstrategie. Konzepte wie SOA Governance, Shared Services (Wiederverwendbarkeit), Change Management sind im Unternehmen etabliert.
  - Die IT-Infrastruktur ist weitgehend auf Basis einer SOA abgebildet.
  - Viele Geschäftsprozesse basieren auf SOA.
  - Herausforderungen sind die Optimierung der SOA-Infrastruktur und Geschäftsprozesse.

3.2 SOA-Projektplanung und Vorgehen

Nach dem Assessment der SOA Readiness, der Überprüfung des Reifegrades für eine SOA-Implementierung, erfolgt die strukturierte Projektplanung, um das für jedes Unternehmen individuelle am besten geeignete Vorgehen zu bestimmen. Ein wesentlicher Punkt in der Planung für die Vorgehensweise von SOA-Projekten ist es, die Unterschiede zwischen Business- und IT-Projekten zu erkennen, geeignete Vorgehensmodelle zu bestimmen sowie weiterführende Methoden zu evaluieren.

Business-Projekte vs. IT-Projekte

Business-Manager sehen die Planung eines SOA-Projektes unter dem Aspekt, dass Unternehmen in der Lage sein müssen, heute und zukünftig Veränderungen schnell, erfolgreich und souverän zu initiieren und zu realisieren. Sie müssen Wege finden, sich intensiver mit Innovationen ihrer Geschäftsprozesse und -modelle beschäftigen zu können.

IT-Manager sehen die Vorgehensweise in einem SOA-Projekt unter dem Aspekt, dass eine SOA die notwendigen Brücken zwischen Geschäftsabläufen und einer flexiblen IT baut und somit Reibungsverluste reduziert. Ein Ergebnis ist eine Kostenreduzierung, welche eine Verlagerung des IT-Budgets von der Maintenance zur Innovation ermöglicht. Darüber hinaus können eine hohe Entwicklungsgeschwindigkeit, eine einfache Integration sowie durch gezielte Wiederverwendung von Services eine schnelle Anpassung der IT-Services an die jeweiligen Anforderungen erreicht werden.

Projektmanager betrachten die Planung von SOA-Projekten hinsichtlich der maßgeblichen Einflüsse auf diese Projekte, die sich aus den jeweiligen Einstiegspunkten in eine Service-orientierte Architektur ergeben:

- Das Geschäftsmodell: Business Process Management als Disziplin erzeugt servicebasierte, transparente Geschäftsprozesse, die flexible Änderungen des Geschäftsmodells unterstützen und eine über SOA Governance kontinuierliche Optimierung der Abläufe ermöglichen.
- Portale: Portale ermöglichen eine rollenbasierte, integrierte Benutzeroberfläche zur Einbindung von Prozessen, Informationen und Abläufen, die eine produktive Zusammenarbeit aller Beteiligten fördert.
- „Information as a Service“ (IaaS) bietet die konsistente Sicht und den Zugriff auf Geschäftsdaten, die Informationen für einen bestimmten Kontext darstellen. Dieser Informationsintegrations-Layer vermittelt zwischen Informations-Bereitstellern und Informations-Abnehmern nach bestimmten, definierten Regeln und transformiert verschiedene Informations-Typen und Formate mit dem Ziel, verschiedene Sichten („Views“) auf Geschäftsdaten für die Weiterverabeitung zu bieten.
- Legacy-Applikationen: Die Modernisierung bestehender Legacy-Anwendungen bietet modulare, wiederverwendbare (Service-orientierte) Anwendungskomponenten, die bestehende IT-Infrastrukturen unterstützen und optimal auf diesen aufsetzen.
- Die Anwendungs-Konnektivität: Die Anwendungs-Interoperabilität und -konnektivität erfordert zwar die Implementierung eines flexiblen Enterprise Service Bus, bietet dafür aber dynamischen, unabhängigen und genormten Zugriff auf die Services eines Unternehmens und erhöht die Werte der Services.
- SOA Governance: SOA Governance definiert und erzeugt effektive Governance-Prozesse und -Verfahren (nicht für IT-Bereiche). Diese führen zu einer stärkeren Bindung der Business- und IT-Komponenten über den gesamten Lebenszyklus und mindern so die „Total Cost of Operation“ und damit auch die „Total Cost of Ownership“.

Wie und wo sollte Ihr Unternehmen ansetzen?

Wie und wo setzt ein Unternehmen am günstigsten an? Für eine SOA-Implementierung gibt es verschiedene Optionen: Eine Service-orientierte Architektur kann erworben, neu aufgebaut oder aus der Entwicklung heraus erstellt werden. Viele Anbieter von Softwarepaketen stellen Services im Rahmen ihrer Produkte bereit. In der Zukunft wird es sogar möglich sein, Services auf Nutzungsbasis von Serviceanbietern zu erwerben, quasi Services zu mieten. Die Möglichkeit, SOA komplett oder auf Nutzungsbasis zu erwerben, ist ein idealer Einstiegspunkt für Firmen mit Standard-Anwendungssystemen oder mittelständische Unternehmen, die sich keine hohen Vorabinvestitionen leisten können. Einige Unternehmen vollziehen den SOA-Einstieg mit der Erstellung einer vollständig neuen Anwendung, andere wiederum entscheiden sich dafür, ihr bestehendes Anwendungsportfolio mit der Zeit umzugestalten.

Unterschiede zwischen Business- und IT-Projekten im Kontext einer SOA

Die wichtigsten Komponenten einer SOA sind ihre Software-Bausteine (Services). Jeder Service steht für eine in sich konsistente und abgeschlossene fachliche Funktion. Ausgehend von prozessorientierten Funktionen stehen bei einer SOA eher die fachlichen als die technischen Aspekte von Geschäftsprozessen im Vordergrund. Erst die technische Interaktionen von Verarbeitungsschritten, die lose gekoppelte Kommunikation von Services, liefert im Rahmen einer SOA das Maß an Agilität und Flexibilität, die notwendig ist, um neue oder geänderte Geschäftsprozesse schnell definieren bzw. neu abbilden zu können.

Im Wesentlichen unterscheiden sich Business-orientierte Projekte von IT-fokussierten Projekten, insofern als sich Business-Anforderungen in der Regel schneller ändern und ändern lassen als IT-Anforderungen. Die kurzfristige Änderung von IT-Anforderungen, oder gar Änderungen der IT-Infrastruktur ist in der Regel aufwendig und mit höheren Folgeänderungen und Abhängigkeiten behaftet als eine Änderung von klassischen Business-Anforderungen. Archiv-Services oder Informations-Services sind hierfür ein passendes Beispiel: Ein Archiv- oder Informationsdienst stellt einem Geschäftsprozess die benötigten Daten über einfache, eindeutig definierte und standardisierte Schnittstellen zur Verfügung. Ein solcher Dienst wird in der Regel nur einmal erstellt, wird aber von mehreren Anwendungen genutzt. Ändert sich nun die Geschäftslogik oder der Geschäftsprozess, bleibt der Archiv- oder Informations-Service meist unverändert, da er bereits als IT-Service standardisiert wurde. IT-Services sind (im Gegensatz zu Business-Services) in der täglichen Praxis aufwendiger zu definieren, sind in der Regel jedoch weniger Änderungen unterworfen

Die folgende Abbildung erläutert die Reichweite einer SOA und die wesentlichen Unterschiede zwischen Business- und IT-Zielen aus Sicht der SOA-Projektaktivitäten :

	Individuelle Web Services	Integrierte Fachfunktion	Unternehmensweite Transformation	Unternehmens-übergreifende Wertschöpfung
Business	Entwicklung von Web Services für eine einfache Applikationsintegration	Integration von Services verschiedener Anwendungen innerhalb einer Fachabteilung	Unternehmensweite Integration von Services verschiedener Anwendungen	Vorbereiten des Unternehmens für externen Austausch: Services bereitstellen oder beziehen und dynamisch kombinieren
IT	Vorbereiten einer Integrationsstruktur	Aufbau einer skalierbaren und verfügbaren Prozess-Laufzeitumgebung	Aufbau einer gemeinsam nutzbaren Integrationsplattform inkl. Servicemanagement	Erweiterung der Infrastruktur, um externe Systeme dynamisch einzubinden

Abbildung 3: Unterschiede zwischen Business- und IT-Zielen aus Sicht der SOA-Projektaktivitäten

SOA-Projekte und deren Einfluss auf das Unternehmen

Die Komplexität von SOA-Projekten resultiert primär daraus, dass eine relativ große Anzahl verschiedener Bereiche (Domains) berücksichtigt und aufeinander abgestimmt werden müssen. Die nachfolgende Abbildung fasst die Domains und ihre Inhalte kurz zusammen.

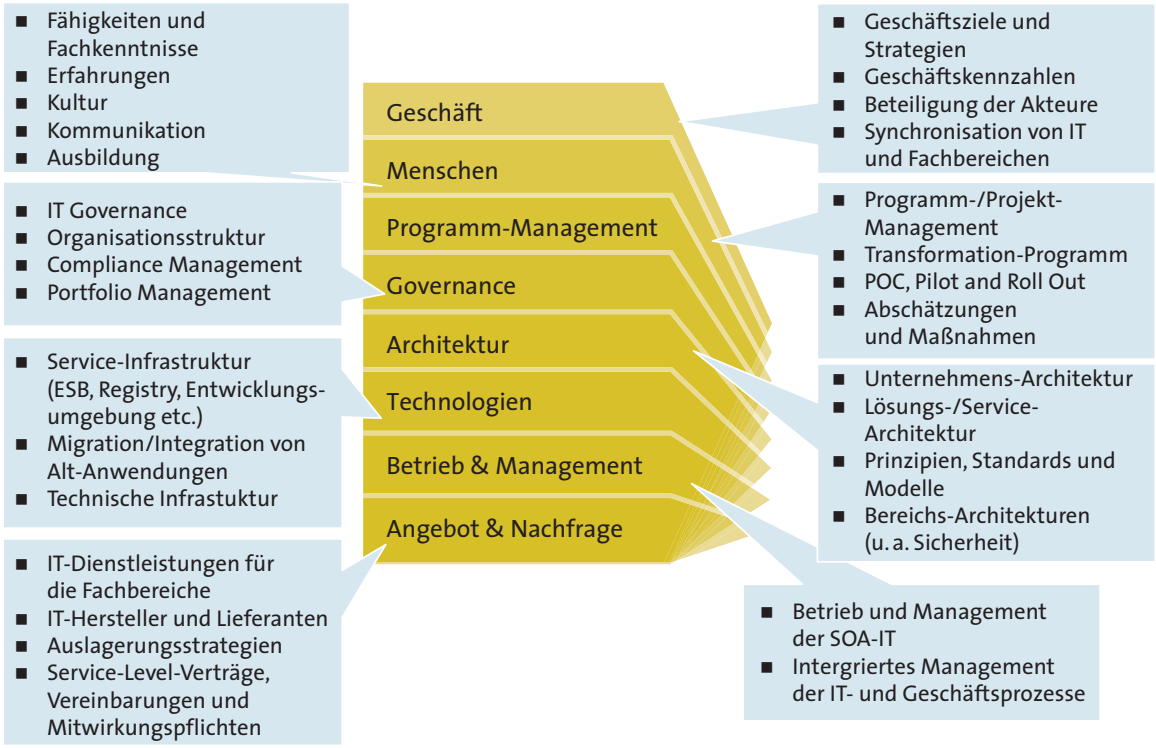


Abbildung 4 : SOA-Domänen-Modell

Die folgenden Herausforderungen gilt es, im Rahmen eines SOA-Projektes zu meistern:

- Schaffung und Governance von SOA Assets (SOA-Services und technische SOA-Anwendungen)
- Definition und unternehmensweite Verankerung von SOA-Prozessen (Prozesse für die SOA-Architektur, Governance und Betrieb sowie SOA Lifecycle Management)
- Management der SOA-Infrastruktur (Flexibilität bzgl. Anwendungs- und Infrastruktur-Plattform)
- Management und Optimierung der IT-Effizienz

Um den Herausforderungen gerecht zu werden, spielt die Kommunikation unter den einzelnen SOA-Disziplinen (Domains) eine zentrale Rolle. Dem Programm- bzw. Projektmanagement obliegt es, diese Kommunikation zu steuern.

Vorgehensmodelle

In früheren SOA-Projekten haben sich zwei klassische, aber gerade in diesem Umfeld sehr konträre Vorgehensmodelle für die Implementierung einer SOA entwickelt:

- der Top-down-Ansatz
- der Bottom-up-Ansatz

Der Analyst IDC Corporation bezeichnete den Top-down-Ansatz als strategisch und den Bottom-up-Ansatz als opportunistisch. In der aktuellen Fachliteratur zu SOA sind beide Vorgehensmodelle bereits ausführlich erörtert worden. Ein Ziel dieses Leitfadens ist es, dem Leser einen Eindruck über die Projekterfahrungen der beteiligten Autoren zu vermitteln.

Der Top-down-Ansatz beschreibt die vollständige, unmittelbare Umstellung der Anwendungen und Systeme eines Unternehmens, also der vollständigen IT, auf eine Service-orientierte Architektur. Dabei werden die Geschäftsprozesse analysiert, neu modelliert und auf Basis einer SOA

auf neuen IT-Komponenten implementiert. Dieser Ansatz bringt neben hohen Einstiegskosten auch ein höheres Risiko mit sich, da zu viele neue Prozesse mit zu vielen neuen Komponenten auch höhere Projektlaufzeiten bedeuten. Eine stark erhöhte Komplexität der Service-Abhängigkeiten und kaum vorhandene Lerneffekte verlängern die Gesamtlaufzeit von SOA-Projekten nach dem Top-down-Ansatz beträchtlich. Längere Projektlaufzeiten beeinflussen das Erreichen von „quick wins", der ROI stellt sich deutlich später ein.

Der Bottom-up-Ansatz besteht in der punktuellen, iterativ geplanten Überführung bestehender Anwendungskomponenten in neue Services. Durch die Anreicherung von Web-Service-Schnittstellen für bestehende Funktionalitäten werden entsprechende Software-Komponenten als neue Dienste zur Verfügung gestellt. Die iterative, „sanfte" Umsetzung der Anwendungslandschaft in neuen Services führt zu unmittelbaren, wenn auch zunächst kleinen Erfolgen, einer schnelleren Erreichung des ROI und einer ausgeprägten Lernphase bei den Projektbeteiligten. Ein SOA-Projekt nach dem Bottom-up-Ansatz beginnt mit der Umsetzung einfacher Geschäftsszenarien, die weder komplexe noch kostenintensive IT-Infrastrukturen erfordern. Typische Beispiele für einen Bottom-up-Ansatz sind die Etablierung von Archiv- und Informationsservices.

Weiterführende Methoden und Vorgehensmodelle zum Aufbau einer SOA:

Service Oriented Architecture Modelling:

Eine Methode zum schrittweisen Aufbau einer SOA, die sowohl die Kernprozesse als auch die differenzierenden Geschäftsprozesse unterstützt.

Component Business Modeling (CBM):

Mit dieser Methode können Unternehmen und deren Geschäftsabläufe in einzelne logische Bausteine zerlegt werden. So werden Kernprobleme identifiziert und adressiert.

Component Infrastructure Roadmap (CIR):

Ein Verfahren, mit dessen Hilfe der derzeitige Stand und Reifegrad der IT beurteilt wird, um die Entwicklung hin zu einer Service-orientierten Architektur zu ermöglichen. Der Ansatz eines SOA-Projektes kann in folgende Schritte strukturiert werden:

- Projektdefinition und Projektziele, ggf. Aufteilung des SOA-Projektes in kleine Einheiten
- Wahl des Projektansatzes (Top-down, Bottom-up)
- Wahl der Technologien
- Management und Monitoring der Projektaktivitäten (Projekt-Governance)
- Generierung von „quick wins" durch gezielte Implementierung von Services, die einen ROI liefern
- Aufbau und Pflege eines umfassend nutzbaren Service Repository
- Gezieltes Monitoring und Steuerung von KPIs und Services (SOA Governance)

Vorgehensmodell bei der SOA-Einführung

- 1. Definition der Projektziele und der Vorgehensweise

Initialer Startpunkt für eine SOA-Analyse ist der SOA Discovery Workshop. In diesem Rahmen identifizieren Unternehmen die richtigen Geschäftsinitiativen für den SOA-Einstieg.

Warum ein SOA Discovery Workshop?

Um den beteiligten Projektmitgliedern einen schnellen Einstieg ohne Risiko und ohne hohe Kosten zu ermöglichen. Darüber hinaus wird so vermieden, dass die Diskussion über eine zukünftige SOA nicht bei Null beginnt.

Der fachliche Einstieg in ein SOA-Projekt sollte mit der Identifizierung von Prozessen beginnen, die in Services umgewandelt werden können. Die Aufwertung bestehender Prozesse durch neue Services hat hier eindeutig Priorität vor dem radikalen Vorgehen, alte Prozesse durch neue zu ersetzen. Dies erfordert jedoch ein Umdenken in der

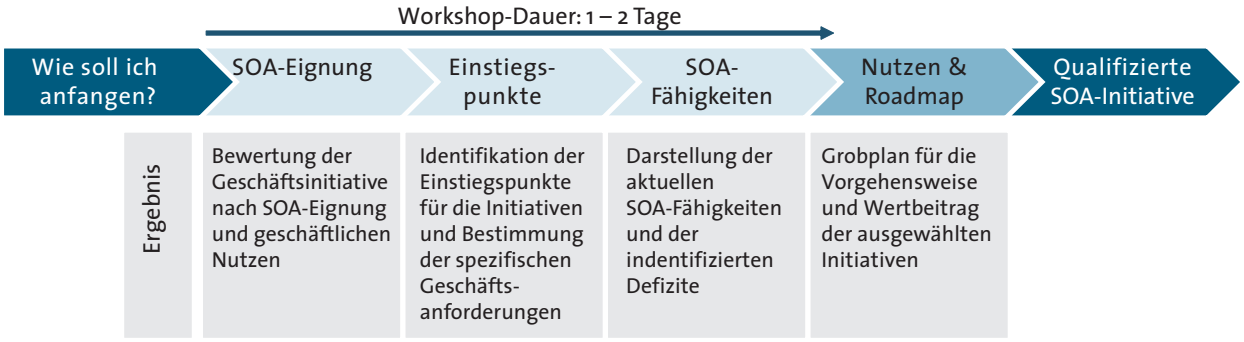


Abbildung 5: SOA Discovery Workshop – Vorgehensweise

Projektdefinition und im Projektmanagement, sowie eine spezielle Vorgehensweise, die die Wiederverwendung von bestehenden Prozessen favorisiert. Als vorteilhaft hat sich folgendes Konzept erwiesen:

- Aufteilung eines großen SOA-Projekts in überschaubare Unterprojekte, die jeweils ihren eigenen Wertschöpfungsbeitrag zum Geschäft leisten und die Basis für weitere SOA-Unterprojekte bilden
- Erarbeitung der Zielarchitektur gemäß den Geschäftszielen und in der Koexistenz der alten und neuen Infrastruktur
- Validierung der Zielarchitektur und des Migrationsprozesses durch die schnelle Realisierung eines Prototyps, welcher die Hauptkomponenten des zukünftigen Systems bereits abbildet. Jede Hauptkomponente besitzt im Prototyp eine begrenzte Funktionalität, ist aber mittels standardisierter Kommunikation kompatibel mit den anderen Hauptkomponenten.
- Schrittweise Erweiterung des Prototyps durch Hinzufügen von definierten Funktionalitäten

Grundsätzlich sollte die Planung eines SOA-Prototyps nur wenige Services beinhalten, um die Komplexität und das Risiko des Projektes nicht zu erhöhen.

- 2. Bestimmung der geeigneten Technologien

Bei diesem Schritt wird auf die Wiederverwendung von bestehenden Technologien („asset reuse“) besonderer Wert gelegt. Es sollen so viele bereits existierende Komponenten wie nur möglich und sinnvoll verwendet werden. Klassische Beispiele sind Server, Datenbanken, Application Server sowie vorhandene Web Services und Protokolle. Müssen Anwendungskomponenten über Web Services und Protokolle vielfältig miteinander kommunizieren, empfiehlt sich der Einsatz eines Enterprise Service Bus (ESB).

- 3. Gezieltes SOA-Know-how aufbauen – beteiligte Gremien und Rollen im SOA-Team

Ein SOA-Projektteam sollte idealerweise so besetzt sein, dass möglichst viele Business- und IT-Infrastruktur-Bereiche abgedeckt werden. In den klassischen Design-Rollen sind Software-Architekten tätig, zu den „Disziplinen“ zählen Enterprise-Architektur, Informationsarchitektur und Anwendungsintegrationsarchitektur. Der Enterprise-Architekt kombiniert geschäftliche Anforderungen zu Services und setzt diese in Dienste um. Der Informations-Architekt konzentriert sich auf die richtige Einbindung von Daten, Datenstrukturen und Metadaten in die zu erstellenden Dienste. Der Anwendungsintegrationsarchitekt ist für die Kommunikation der Dienste untereinander, die technische (Wieder-)Verwendbarkeit von Diensten und die Konformität einer SOA Registry bzw. eines Repository verantwortlich.

Ein ausgesprochener Zusammenhang der Services untereinander ist für den Softwareentwickler zunächst nicht erkennbar. Er muss wissen, was die jeweiligen Services leisten sollen, mit wem diese kommunizieren und

welche SLAs der jeweilige Service erfüllen soll. Aufgrund der vielfältigen Spezialisierungen ist die Teamzusammenstellung für SOA-Projekte besonders wichtig. Das Projektteam sollte idealerweise aus einer Kombination von IT-fokussierten Rollen wie Entwicklern, Software-Architekten, Projektleitern und businessbezogene Rollen wie Business-Analysten (für die Geschäftsprozess-Modellierung), Quality-Assurance-Analysten bzw. KPI-Analysten bestehen. Speziell bei der Entwicklung von Services arbeiten Business-Analysten Hand in Hand mit den SOA-Entwicklern („Integration Developer"), die die Geschäftsprozesse in eine Modellsprache (z. B. BPEL) umsetzen und die Integration bzw. Abläufe der Prozesse in der Prozess-Engine festlegen. In einer SOA werden Dienste und Abläufe flexibel gestaltet. Der Business-Analyst und die SOA-Entwickler haben dabei die Möglichkeit, die Dienste und Abläufe zu simulieren und je nach Anforderung bzw. Ablauf flexibel zusammenzustellen, um die Qualität der Serviceabläufe bereits in der Integrationsphase zu verbessern. Das primäre Ziel sollte in jedem Fall die multiple Wiederverwendung der Services sein.

Wie bereits dargestellt erfordern SOA-Projekte ein breites IT-Wissen in vielen Disziplinen. Neben den klassischen Fachabteilungen müssen insbesondere die vielen IT-Bereiche einbezogen werden. Spezialisten zu Themen wie Portale, Identity/Access Management, Datenintegration/Datenversorgung und Service Monitoring sind in Projekten nach dem Bottom-up-Ansatz intensiv beteiligt.

- 4. Sukzessive Annäherung an die Ziel-Architektur

Der am Bottom-up-Modell orientierte Ansatz fordert die Definition und Erstellung von Service-Komponenten bzw. die sukzessive Erweiterung vorhandener Servicekomponenten. Diese Entwicklung bzw. Erweiterung wird aus der IT-Perspektive unter der Berücksichtigung der Businessanforderungen heraus getrieben.

- 5. Projekt-Governance

Während aller Phasen eines Projektes stellt das SOA-Governance-Team die Abstimmung zwischen den Projekten und die Einhaltung der in der Planungsphase definierten Standards sicher.

- 6. Aufbau einer Service Registry und eines Service Repository

Das Service Repository und die Registry sind sehr wichtige Elemente einer SOA.

Das Metadaten-Repository, das das leistungsfähige Framework und die Erweiterbarkeit aufweist, um den individuellen Anforderungen des unterschiedlichen Serviceeinsatzes gerecht zu werden, sollte möglichst früh in einem Projekt aufgesetzt werden.

Das Business Services Repository speichert Metadaten, die zur Erstellung von Business-Services in SOA-Systemen verwendet werden. Die Metadaten beschreiben die Services, Richtlinien und Beziehungen untereinander sowie die Berechtigungen für Abonnenten. In der Service Registry sind die Process-Services hinterlegt, die Informationen zu den Services enthalten (z. B. zur Serviceschnittstelle, zu deren Prozessen und zu deren Parametern).

Es ist wichtig, in der Service Registry und Repository alle Bereiche einer SOA abzudecken. Dazu gehören Prozessbeschreibungen, Policies, Business-Rules und die Services selbst. Um eine volle Transparenz zu erhalten, muss beschrieben werden, wie die verschiedenen Artefakte einer SOA zusammenhängen. Des Weiteren empfiehlt sich im Rahmen der SOA Governance die Einführung eines Service-Informations-Management (SIM).
Ein Service-Informations-Management ist ein mit Metadaten angereichertes Informationssystem, welches die Strukturen, Abhängigkeiten und Definitionen von Services aus fachlicher und technischer Sicht zentral zusammenfasst und modellgestützt für eine erfolgreiche SOA Governance darstellt.

- 7. SOA Governance und Business Activity Monitoring einführen

Die Abschnitte 3.6 und 5.2 zum Thema SOA Governance beschreibt das entsprechende Vorgehen.

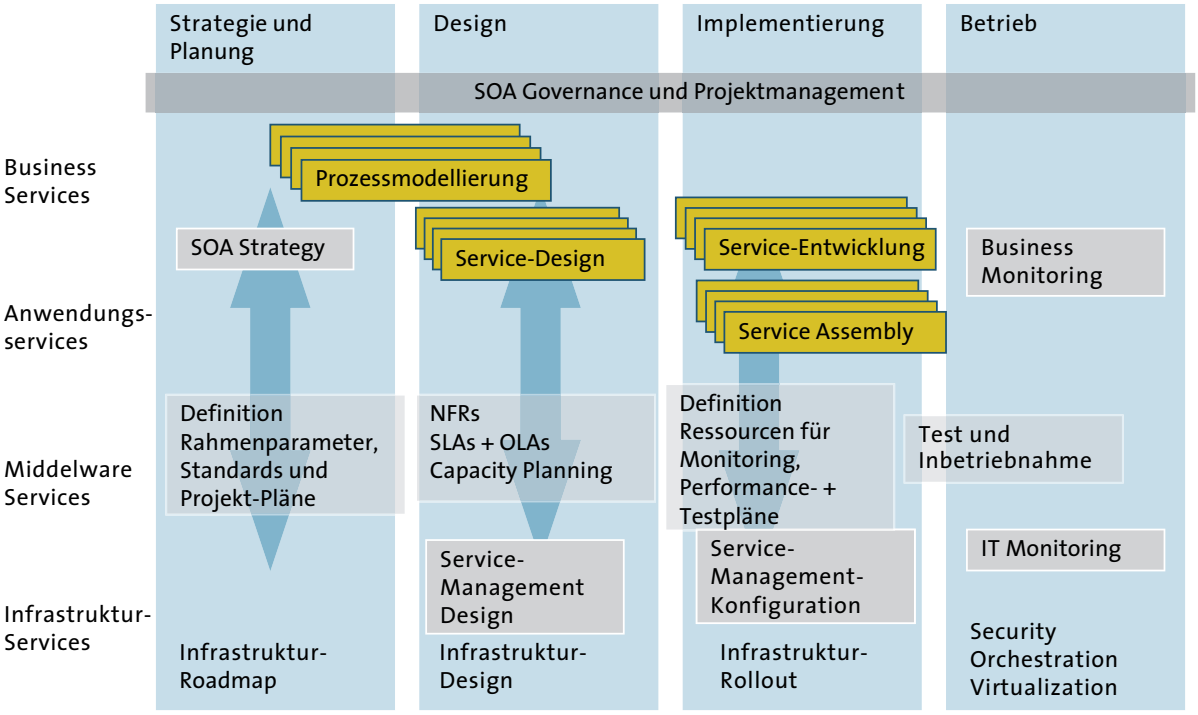


Abbildung 6: SOA Governance und Projektmanagement in den einzelnen Phasen

Fazit

- Es gibt verschiedene Wege und Szenarien zur Einführung einer SOA.
- In der Praxis hat es sich bewährt, zunächst einige wenige Use Cases (Funktionen) zu verwirklichen und die Gesamtinfrastruktur aufzusetzen
- SOA Governance ist eine Projektaufgabe mit höchster Priorität – es muss eine Verbindung zwischen wiederverwendbaren Services und wiederverwendbaren Fachfunktionen bestehen.
- Das notwendige Know-how zur Einführung einer SOA lässt sich gut im Rahmen eines Prototyps sammeln.
- SOA-Entwickler müssen sich darauf konzentrieren bestehende Services gezielt wiederzuverwenden.
- Die Innovationszyklen müssen auf Anwendungs- und Infrastrukturebene zusammenpassen.
- Selbst in SOA-Einstiegsprojekten werden auf der Business- und der IT-Seite mehrere Rollen und Spezialfähigkeiten benötigt.

3.3 Messung des Return on Investment

Return on Investment – Wodurch rechtfertigen sich die hohen Kosten?

Kein Unternehmen wird seine Software-Infrastruktur in wenigen Jahren vollständig auf SOA umstellen können. Jedoch stehen aufgrund der übertriebenen Darstellungen in den Medien große Erwartungen im Raum, die das technisch Mögliche und vor allem wirtschaftlich Sinnvolle bei weitem übertreffen. Hier gilt es also, einen Rahmen zu definieren, auf den sich alle SOA-Aktivitäten beziehen. Dieser Rahmen beschreibt zunächst die Motivation für die teilweise beträchtlichen Investitionen und den Plan für eine konkrete Umsetzung. In diesem Kapitel wird eine strukturierte Herangehensweise für eine Kosten-Nutzen-Bestimmung (ROI) vorgestellt, die einen notwendigen Brückenschlag zwischen der technisch orientierten Welt der Informatik und der kaufmännischen Realität des Managements darstellt.

Die Berechnung des ROI ist nur bedingt quantitativ möglich. Daher sollte die Vorgehensweise auch qualitative Werte wie Flexibilität und Effizienz aber auch verschiedene Risiken berücksichtigen. Eine schrittweise (inkrementelle) Annäherung an ein glaubwürdiges Ergebnis beginnt mit der Analyse der wesentlichen Treiber, also der Motivation, auf deren Basis eine SOA erwachsen soll. Die Analyse berücksichtigt alle Interessenvertreter sowie die organisatorischen, firmenpolitischen und gesetzlichen Rahmenbedingungen. Der konkrete Umsetzungsplan zeigt alle Maßnahmen auf, die hinsichtlich ihres Wertes qualitativ oder quantitativ abgeschätzt werden können. Unterschiedliche Sichtweisen auf dem Weg zu einem quantitativen Ergebnis ermöglichen eine mehrfache Rückmeldung (Iterationen) auf bereits erfolgte Abschätzungen. Das Ziel ist letztlich eine Aussage über den Zeitpunkt, ab dem die Investitionen einen Gewinn für zukünftige Projekte erbringen.

Vorgehensweise

Der Weg zu einer funktionierenden SOA ist recht lang. Der Nutzen einer solchen Architektur wird erst nach einiger Zeit sichtbar. Unternehmen stehen vor der schwierigen Aufgabe, die Investitionen in einen Umstieg nicht nur durch intuitiv verständliche Vorteile, sondern durch einen greifbaren und quantifizierbaren Nutzen zu rechtfertigen. Neben den initialen Aufwendungen für Software, Schulungen und Umstrukturierungen erfordert eine Messung des Return on Investment (ROI) typischerweise eine längere Phase der Stabilisierung sowie den erfolgreichen Abschluss erster Projekte. Zudem haben SOA-Implementierungen immer auch einen fachbereichübergreifenden Charakter, welcher der traditionellen Vergabe von Projektbudgets und ROI-Messungen entgegensteht. SOA-Implementierungen sind einem ständigen Wandel unterworfen: Häufige Änderungen und Erweiterungen müssen möglich sein. Dieses Ziel macht in vielen Umgebungen eine Kosten-/Nutzen-Rechnung unmöglich. Wie lässt sich also eine dauerhafte Unterstützung durch das Unternehmensmanagement erreichen?

Vor allen Dingen muss ein Unternehmen erkennen, dass Investitionen in die IT nicht nur Technologien, sondern auch die Geschäftspraktiken verändern. Die Ausrichtung der IT am Geschäft (Business Alignment) und das Verständnis für Veränderungen am Markt werden zum wesentlichen Erfolgsfaktor. Der ROI basiert also nicht auf Anschaffungen, sondern insbesondere auf veränderten Arbeitsweisen und organisatorischen Strukturen. Es werden Entscheidungsgremien gegründet, die auf Basis eines Portfolio-Managements die IT-Investitionen bewerten. Dabei erweisen sich einfache Modelle wie der traditionelle ROI nach dem Du-Pont-Schema oder andere Finanzmetriken als nicht ausreichend, da sie eher an der Vergangenheit orientiert sind und zukünftige Entwicklungen und Risiken nicht berücksichtigen. Sie unterschlagen häufig die nicht monetären Gewinne sowie begünstigende Auswirkungen auf zukünftige Projekte [SYM 2006]. Daneben sind manche Projekte nicht finanziell motiviert, haben aber trotzdem große Auswirkungen auf die geschäftlichen Möglichkeiten des Unternehmens: Dabei kann es sich um die Reaktion auf eine Bedrohung durch die Konkurrenz oder die Antwort auf Regularien des Gesetzgebers (z. B. Sarbanes-Oxley) handeln. Besser geeignet sind daher Verfahren wie Business Value Index (BVI), Total Economic Impact (TEI), Val IT oder Applied Information Economics (AIE). TEI wird dem Charakter von SOA-Entwicklungen besonders gut gerecht: Schwerpunkte bilden hier die erwartete Flexibilität bei zukünftigen Projekten und die Analyse der Risiken im Hinblick auf veränderte Rahmenbedingungen in der Zukunft.

Business Value Index

Die BVI-Methodologie wurde 2001 von der Firma Intel entwickelt, um die Wertschöpfung ihrer IT-Projekte zu messen. Hier werden auch nicht monetäre Vorteile berücksichtigt und die Effizienzverbesserung der IT-Organisation wird kalkuliert. BVI ist eine praxiserprobte Methode, die in Form von Dokumentation frei verfügbar ist. Sie wird nicht durch Beratung seitens Intel unterstützt.

Total Economic Impact

TEI ist der Ansatz von Forrester Research und legt gegenüber anderen Methoden einen größeren Wert auf die gewonnene Flexibilität und das mögliche Risiko eines Projektvorhabens. TEI ist nicht frei verfügbar. Zeit und Geld müssen aufgewendet werden, um die Methode zu lernen und ihre Werkzeuge anzuwenden.

Val IT

Val IT stellt eine Erweiterung des COBIT Governance Framework des IT Governance Institutes (ITGI) dar. Der inhaltliche Schwerpunkt liegt hier auf dem Portfolio-Management und verfolgt einen eher quantitativen Ansatz. Die Methode ist relativ neu, eng mit dem COBIT-Framework verknüpft und hat daher für Bestandsnutzer von COBIT einen besonderen Wert.

Applied Information Economics

AIE ist ein 10 Jahre altes Verfahren zur Bewertung von IT-Investitionen. Es ist stark quantitativ ausgerichtet und basiert auf bekannten statistischen Methoden zur Analyse von Risiko und Gewinn. Es ist nicht weit verbreitet, wird aber einem gesteigerten Interesse an Risikoanalyse gerecht.

Eines dieser Bewertungsmodelle oder ein speziell auf das eigene Unternehmen angepasstes dient nun als Werkzeug im Rahmen einer strukturierten Vorgehensweise. Das Modell wird im Einzelnen durch Bewertungsmaßstäbe ergänzt, um die nicht monetären Effekte einer SOA abzuschätzen. Aufgrund der Komplexität dieser Aufgabe kann im Gegensatz zum Umgang mit Balanced Scorecards (BSC) [KAN 1997] auch auf konkrete Kennzahlen verzichtet werden, stattdessen werden generische Maßstäbe verwendet. Die im Folgenden vorgestellte Vorgehensweise kombiniert die konkret quantifizierbaren Aspekte mit der subjektiven bzw. auf empirischem Wissen basierenden Beurteilung von Vorteilen und Risiken. Sie schreibt vier Phasen vor, die iterativ durchlaufen werden, um neu gewonnene Erkenntnisse in die Verbesserung der Abschätzungen einfließen zu lassen.

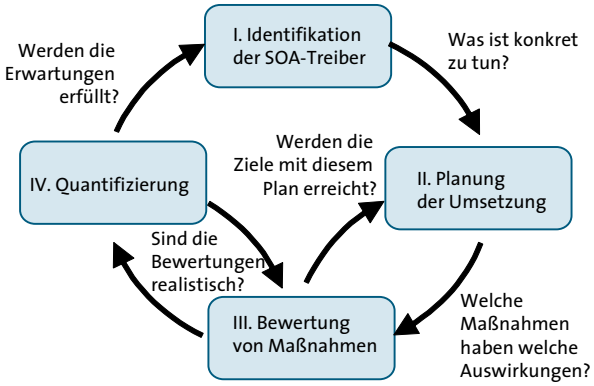


Abbildung 7: Die einzelnen Phasen werden iterativ durchlaufen

In der ersten Phase werden die wesentlichen Treiber und Erwartungen in eine SOA untersucht. Ist die Veränderung durch einen „Leidensdruck“ motiviert oder soll das Unternehmen für zukünftige Veränderungen gewappnet werden? Mit diesen Ergebnissen startet der Business-Plan für die SOA-Initiative. Die zweite Phase wird zunächst nur grob ausgefüllt, indem die wesentlichen Investitionen und die zeitliche Abfolge von Einführungen, Umstrukturierungen usw. geplant werden. Diese Überlegungen können als Einstieg in die Feinplanung und Bewertung von Maßnahmen dienen. Sie können auch parallel zur dritten Phase untersucht werden. Dazu gehört auch die Analyse früherer Budgets und eine quantitative Untersuchung der IT-Prozesse, die es mit einer SOA zu verbessern gilt. Die konkreten Maßnahmen werden nun auf die durch sie provozierten Veränderungen bzw. Verbesserungen hin bewertet. Hierbei findet eine Rückmeldung und eine Verbesserung des zuvor grob erstellten Plans statt. Ist man sich über die konkreten Ziele und ihre Umsetzung klar, so können diese quantitativ erfasst werden. Der Versuch auch subjektive Verbesserungen quantitativ auszudrücken, führt wiederum zu einer Korrektur der Bewertungen. Die verschiedenen Phasen bieten unterschiedliche Perspektiven auf eine SOA-Einführung und verbessern so die Abschätzungen des Return on Investment.

Phase I – Identifikation der wesentlichen SOA-Treiber

Eine ROI-Untersuchung wird Tom DeMarcos Prinzip gerecht, dass sich nur beherrschen lässt, was konkret bekannt, d. h. messbar ist: „You cannot manage what you cannot measure“. Dieses Prinzip, das den Balanced Scorecards zugrunde liegt, wurde in den 1990er-Jahren von Kaplan and Norton um die Erkenntnis erweitert, dass nur gemessen werden kann, worüber es einen konkreten inhaltlichen Überblick gibt: „You cannot measure what you cannot describe“. Zur abstrakten Beschreibung eines Unternehmens und verschiedener Perspektiven auf seine Wertschöpfungsketten entwickelten sie daher den Ansatz der Strategy Maps [KAN 2004]. Mit diesen lässt sich transparent machen, welche Strategien in einem Unternehmen verfolgt werden (sollten) und welche Prioritäten dafür gelten. Die Strategy Maps bieten auch eine sehr gute Grundlage für die Analyse der SOA-Treiber und helfen darüber hinaus weniger offensichtliche Treiber aufzudecken. Ein weiterer Vorteil liegt in der Identifizierung der entsprechenden Interessenvertreter (Stakeholder), deren unterschiedliche Wünsche und Ängste jeweils berücksichtigt werden müssen.

Strategy Maps

Der Strategy-Maps-Ansatz von Kaplan und Norton basiert auf der Existenz einer Balanced Scorecard. Verschiedene Perspektiven eines Unternehmens werden hierbei mit IT-gestützten Prozessen in Verbindung gebracht. Sie werden eingeteilt in sog. interne Perspektiven, wie Produktion, Kundenmanagement und Innovationsprozesse. Des Weiteren gibt es die Dimension „Lernen und Entwicklung“, zu der Finanzen, Personal und Strategien zählen. Zum Grundverständnis dieses Modells zählt die enge Verzahnung von IT und Business (Alignment), die nur in Übereinstimmung einen wirtschaftlichen Beitrag der IT zum Unternehmenserfolg leisten.
Das macht die Anwendung dieses Ansatzes für IT-Strategien besonders wertvoll.

Wird in einem Unternehmen nicht bereits mit Balanced Scorecards gearbeitet, so wird der Einsatz von Strategy Maps ein sehr aufwändiges Unterfangen sein. In diesem Fall kann ein pragmatisches Modell für die Analyse und Kategorisierung von SOA-Treibern sowie der Klärung der zentralen Fragen angewendet werden.

Der SOA-Entscheidungsprozess wird in der Regel durch einen technischen oder wirtschaftlichen Leidensdruck angestoßen; oft hängen beide eng zusammen. Eine wichtige Motivation sind die hohen Wartungskosten proprietärer Anwendungen (wirtschaftlicher Leidensdruck), die meist begleitet werden von einer geringen Flexibilität bei Änderungsanforderungen (technischer Leidensdruck). Neben diesen unternehmensinternen Treibern existieren aber auch externe Treiber, beispielsweise durch gesetzliche Vorgaben wie Basel II oder den Sarbanes-Oxley Act, welche die Unternehmen zu einem Umbau ihrer IT-Umgebung zwingen.

Wie bereits erwähnt sind bei der Ausarbeitung einer Mehrwertanalyse auch die entsprechenden Interessenvertreter zu berücksichtigen. Externe Treiber wie Marktveränderungen oder gesetzliche Regelungen werden eher die kaufmännische Seite, also das Management, zum Handeln motivieren. Interne oder technische Treiber dagegen werden zunächst von den IT-Verantwortlichen wahrgenommen. Erfahrungsgemäß gehen SOA-Initiativen meist von den IT-Abteilungen aus, was häufig zu Schwierigkeiten in der Kommunikation und Rechtfertigung von Budgets führt. Nur mit einer konsequenten Ausrichtung der IT an den geschäftlichen Erfordernissen des Unternehmens ist mit einer dauerhaften Unterstützung durch das Management zu rechnen, was für die strategische Komponente der SOA auch zwingend notwendig ist (Beispiel: abteilungsübergreifende strategische Budgets).

Um eine SOA-Initiative zu rechtfertigen, lohnt sich eine Kategorisierung der Nutzenargumente. So lässt sich eine vollständigere Abdeckung aller wichtigen Argumente sicherstellen und ermöglicht, gezielt die unterschiedlichen beteiligten Parteien anzusprechen. Eine Grundlage für eine solche Kategorisierung stellt die folgende Liste dar:

Business-Innovation
- Schnellere Umsetzung von Fusionen und Übernahmen
- Neue Produkte und Angebote durch Neubündelung vorhandener Leistungen
- Verbesserte Kundenzufriedenheit durch innovative Angebote

Anwenderproduktivität
- Gesteigerte Produktivität durch intuitive Benutzeroberflächen auf Basis von AJAX führen zu mehr Transaktionen pro Zeiteinheit
- Reduzierter Trainingsaufwand
- Weniger operative Fehler durch Vermeidung manueller Datenübertragung

Prozesseffizienz
- Gesteigerte Prozessautomatisierung
- Verbesserte interne Prozesse, wie zum Beispiel für Compliance und Genehmigungsverfahren
- Verbesserte externe Prozesse, beispielsweise im Vertrieb oder im Einkauf

IT-Projekte beschleunigen
- Wiederverwendung vorhandener Funktionen
- Redundante IT-Anwendungen verringern
- Schnellere Reaktion auf neue Anforderungen aus den Geschäftsbereichen

Betriebskosten
- Geringerer Wartungsaufwand der Eigenentwicklungen
- Vereinheitlichung der Infrastruktur (Produkte und Konzepte)
- Bessere Nutzung von Ressourcen

Eine SOA kann auch in unterschiedlichen Reifegraden (Maturity Model) umgesetzt werden (siehe Abschnitt 3.1). Je nach den Erfordernissen des jeweiligen Unternehmens kann eine einheitliche Infrastruktur auf Basis eines Enterprise Service Bus oder ein fortgeschrittenes Business Activity Monitoring (BAM) Ziel der Aktivitäten sein. Eine Zuordnung unterschiedlicher SOA-Reifegrade zu den oben genannten Kategorien hilft zum besseren Verständnis der verschiedenen Interessenslagen und zur Fokussierung auf die wichtigsten Ziele.

Sobald die wesentlichen Treiber und Ziele für eine SOA formuliert sind, kann sich im nächsten Schritt die Planung der Umsetzung darauf aufbauen.

Phase II – Planung der Umsetzung

Im ersten Moment scheint die konkrete Planung von Produkten, Organisationsstrukturen, Umbauten usw. vor der Bewertung und Priorisierung der Ziele etwas verfrüht. Doch ist gerade eine Priorisierung schwierig, solange man das Vorhaben nur abstrakt betrachtet. Mit der Planung der praktischen Umsetzung wird nach einer langen Phase theoretischer Arbeit wieder an „Bodenhaftung“ gewonnen. Da die notwendigen Veränderungen und Investitionen größtenteils offensichtlich sind, lässt sich nun ein erster Plan aufstellen. Die vorgestellten Phasen werden iterativ durchlaufen, so ist eine Korrektur wiederholt möglich. Im Übrigen können diese Aktivitäten teilweise auch parallel abgearbeitet werden.

Wie sieht also die Strategie für die Einführung einer SOA aus? Welche Aspekte einer SOA werden wann von wem mit welchen Produkten und in welchem Zeitraum umgesetzt? Die zentralen Themen in dieser Phase sind die Kosten für Anschaffungen, Umbauten und Ausbildung sowie der Zeitplan, nach dem die Umsetzung durchgeführt werden soll:
- Einmalige Investitionskosten, Lizenzen über mehrere Jahre, Personal
- Zeitplan für die Budgetierung der unterschiedlichen Kosten
- Durchschnittliche Projektkosten (Entwicklung und Wartung

Der erste Punkt ist kritisch: Zum einen beinhaltet er zusätzliche jährliche Lizenzen, Wartungsverträge usw., die nicht zu den Abschreibungen gezählt werden können, und zum anderen führt er Personalkosten auf, die schwierig zu planen sind. Es ist im Einzelfall zu prüfen, ob die Personalkosten überhaupt für die SOA-Berechnung relevant

sind. In Bezug auf die laufenden Kosten für Lizenzen u. ä. muss eine Gegenüberstellung zu den aktuellen Kosten erfolgen. Mit einem Betrachtungszeitraum für diese Planwerte von drei bis fünf Jahren trägt man der langfristigen Wertschöpfung der SOA Rechnung.

Der Plan für die Einführung einer SOA muss auf mehrere Jahre und inkrementellen Ausbau ausgelegt sein. In Ergänzung zu den langfristigen Zielen sollte der ROI auch durch ein „gesundes" Maß an kurz- und mittelfristig abzuschließenden Projekten gestützt werden. Daher

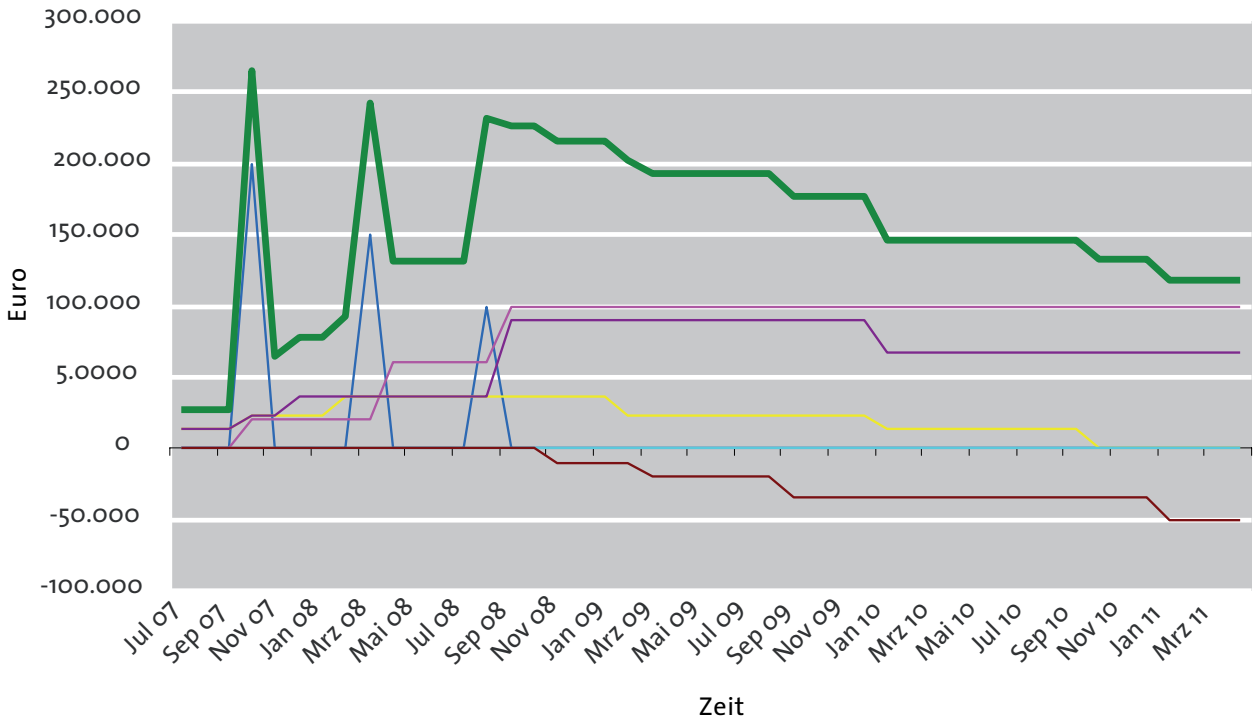


Abbildung 8: Investitionen, Abschreibungen, Personal, Einsparungen,Umbauten usw.

ist es empfehlenswert, mit Fragen der technologischen Infrastruktur zu beginnen (MOM, ESB) und erste Erfahrungen damit zu sammeln. Gleichzeitig kann die Planung der Unternehmensarchitektur (Enterprise Architecture) konkretisiert werden. In Abhängigkeit des bereits erreichten SOA-Reifegrads wird sich dieser Plan ändern und andere Schwerpunkte verfolgen.

Die Vergabe von Budgets ist in der Regel an einzelne Entwicklungsprojekte gekoppelt – genau dort wirken sich auch die meisten Vorteile einer SOA aus. So lohnt es sich, die Projektkosten detailliert aufzuschlüsseln. Die erwarteten Vorteile, die bereits in Phase I aufgelistet und kategorisiert wurden, werden später in Phase IV auf die Projektkosten abgebildet und mit entsprechenden Gewichtungen versehen. In den seltensten Fällen wird der Betrieb einer Anwendung in das Budget eines Entwicklungsprojekts eingeplant; wahrscheinlich wird man die Betriebsaufwände eher in der obigen Kostenkurve (Investitionen, Abschreibungen, Personal, Einsparungen, Umbauten usw.) berücksichtigen. Im folgenden Beispiel sind die Betriebskosten innerhalb des ersten Jahres berücksichtigt.

Beschreibung von Kostenfaktoren		Anteil [%]
Service-Benutzung	Front-End-Anwendungen neu entwickeln	5
	Front-End-Anwendungen verändern	5
	Ausbildung Service-Verwendung	1
Service- Infrastruktur	Infrastruktur-Software (Lizenzen auf 1 Jahr)	6
	Software-Wartungsverträge (auf 1 Jahr gesehen)	3
	Ausbildung Service-Infrastrukturaufbau	1
	Aufwand für Infrastukturaufbau	8
	Integration und Adaption	10
	Systems Management	3
	Hardware (Anschaffung)	5
	Betrieb erstes Jahr (Harware, Netze, Admin, ...)	4
	Externe Beratung: Technik (Spezialisten)	2
	Centre of Excelence CoE und Governance	0
Service-Entwicklung und -Wartung	Neue Anwendungen entwickeln	15
	Änderungen und Erweiterungen der Services	14
	Anwendungen mit Schnittstellen kapseln	5
	Ausbildung Service-orientierte Softwareentwicklung	1
	Externe Beratung: Methoden (Spezialisten)	1
	Fehlerbehebung	4
	Security	3
	Logging, Auditing	2
	Filtering, Routing, Transformation usw.	2
	Summe [%]	100

Die Kostenfaktoren sind nun grob ermittelt und ein erster Fahrplan für die Erreichung der in Phase I definierten Ziele ist erstellt. Die in dieser Phase ermittelten Werten werden nach Feedback aus den Phasen III und IV noch korrigiert (iterativ). Im nächsten Schritt können detaillierte Maßnahmen ausgearbeitet werden, die dem Fahrplan entsprechen. Damit erfolgt eine Bewertung und erste „weiche" Quantifizierung. Die Kosten werden in der folgenden Phase erneut aus einem anderen Blickwinkel betrachtet, nämlich im Hinblick auf die Veränderungen durch die geplanten Maßnahmen.

Phase III – Bewertung von Maßnahmen

In Phase I wurden bereits die SOA-Treiber, d. h. die Motivation für die SOA-Einführung, genannt. Im Rahmen der Kostenbetrachtung ist ein grober Plan entstanden, der sich an den üblichen Faktoren einer SOA-Infrastruktur orientiert. Die Einführung eines Enterprise Service Bus, einer Registry und vielleicht auch einer BPEL Engine sind grundlegende Maßnahmen, die von notwendigen Ausbildungsmaßnahmen und Pilotprojekten begleitet werden. Je nach organisatorischer und politischer Umgebung werden diese Maßnahmen nun aufgeschlüsselt

und als zielführende Aktivitäten präsentiert – gegebenenfalls mit Bezug auf konkrete Abteilungen, die in der Vergangenheit ähnliche Aufgaben erledigt haben. Die organisatorischen und politischen Zusammenhänge dürften in der ersten Phase transparent geworden sein. Aufgaben wie IT-Governance, IT-Architektur, Lieferantenmanagement, Monitoring von Geschäftsprozessen usw. gewinnen in einer SOA andere – teilweise deutlichere – Profile. Die nun betrachteten Maßnahmen haben oftmals den Charakter von erwarteten Zielen, die mit der SOA verfolgt werden. Sie machen deutlich, was sich mit einer SOA verbessern soll und welchen Beitrag die konkrete Maßnahme dazu beisteuert. Beispiele sind:

- Vereinheitlichung von Middleware-Technologie auf Basis eines ESB
- Erhöhen der Datenqualität durch ein Business Object Model (BOM)
- Steigerung der Service-Wiederverwendung durch kontrolliertes Service-Design
- Verbesserung der internen Prozesse für Compliance und Genehmigungsverfahren
- Bereitstellung von Kernfunktionalität des Legacy-Systems XY über Standardschnittstellen
- Vereinfachen der internen Kommunikation durch Etablierung von Service Domains
- Beschleunigung von Inbetriebnahmen auf Basis von UDDI 3.0-Funktionalität

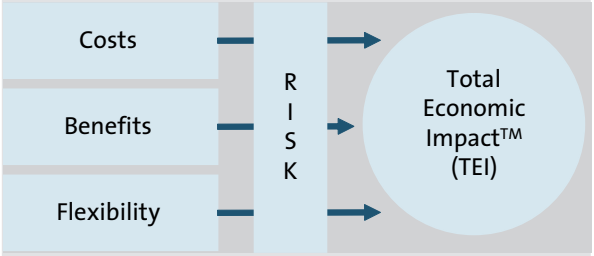
Bei der Beschreibung und Bewertung der Maßnahmen wird jeweils der Istzustand dem Sollzustand bzw. der erwarteten (verbesserten) Situation gegenübergestellt. Im einführenden Abschnitt wurde bereits auf das Verfahren „Total Economic Impact“ (TEI) von Forrester Research hingewiesen. Angelehnt an die dort definierten Kriterien werden Kosten, Vorteile, Flexibilität und Risiken jeder einzelnen Maßnahme betrachtet.

Total Economic Impact (TEI)

TEI ist ein von Giga Research entwickeltes Modell zur Aufnahme und Kommunikation des Mehrwerts bzw. ROI von IT-Initiativen in allgemeiner betriebswirtschaftlicher Sprache. Dabei werden die folgenden vier Kriterien berücksichtigt:

- Geschäftliche Vorteile
- IT- oder Projektkosten
- Flexibilität und zukünftige Optionen
- Risiko und Unsicherheit

Die Abbildung zeigt das prinzipielle Konzept: Kosten, erwartete Vorteile und Flexibilität werden beziffert, mit einer Risikoanalyse kombiniert und daraus der erwartete ROI berechnet.



Quelle: Giga Research, eine Tochtergesellschaft der Forrester Research, Inc.

Die geschäftlichen Vorteile umfassen Produktivität, effektive Auswirkungen auf den Marktanteil, organisatorische Effizienz und Kundenzufriedenheit im Sinne von zusätzlichem Umsatz.

Die zu untersuchenden Kosten beziehen sich auf Hard- und Software (einmalige und Update-Kosten), Wartung und Betrieb von Technologie (Personal und Werkzeuge).

Die erwartete Flexibilität drückt sich in Einsparungen innerhalb von Folgeprojekten aus. Diese sind eventuell noch nicht budgetiert. Die Option einer gesteigerten zukünftigen Flexibilität stellt dennoch einen Wert dar. Keine Chance ohne Risiko und keine Vermeidung von Veränderungen ohne Risiko: Risiko drückt die Unsicherheit über das erwartete Eintreffen von Annahmen aus. Die erwarteten Vorteile werden durch mögliche Risiken geschmälert. Das Risiko in Bezug auf die Kosten hängt beispielsweise ab von der Projektgröße, von Technologien, Produktherstellern, verfügbarem Know-how und anderen Rahmenbedingungen. Auch in Bezug auf die erwarteten geschäftlichen Vorteile existieren Risiken, z. B. in Form von sich ändernden Strategien des Managements, von Veränderungen im Markt oder Akzeptanz der betroffenen Benutzer.

Das TEI-Modell wird in diesem Kontext nur in seinen Grundzügen angewendet. Die oben genannten Kriterien werden um passende Bewertungsmaßstäbe ergänzt, anhand derer jeweils der Istzustand mit dem Sollzustand (nach durchgeführten SOA-bezogenen Maßnahmen) verglichen. Die Bewertung der Kosten ist am einfachsten: Der Istzustand entspricht 100 % der üblichen Kosten, der Sollzustand entspricht einem entsprechend der

definierten Erwartung veränderten Prozentwert – in der Regel ist dieser niedriger. Die Maßstäbe für die Kriterien Vorteile, Flexibilität und Risiken sind ungleich schwieriger zu definieren.

Für den Wertebereich der Schätzungen – denn es handelt sich hinsichtlich der Kriterien „Vorteile", „Flexibilität" und „Risiken" um Schätzwerte – eignen sich besonders die diskreten Werte der Fibonacci-Folge* oder anderer nicht linearer Folgen, da mit ihnen Relationen und statistisch seltenere Extremwerte angemessen ausgedrückt werden können. „Vorteile" lassen sich beispielsweise mit folgenden Effizienzfaktoren ausdrücken:

ID	Wertebereich für „Vorteile"
1	Kaum spürbare Veränderung gegenüber dem Durchschnitt
2	Prozesse laufen etwas flüssiger, man spürt eine leichte Optimierung
3	Es macht sich eine deutlich höhere Produktivität bemerkbar.
5	Die Effizienz ist verdoppelt, die Fehlerrate stark verringert.
8	Die gleichen Ziele werden um ein Vielfaches schneller erreicht.
13	Die Effizienz, d. h. Geschwindigkeit, ist verzehnfacht; Fehler kommen praktisch nicht mehr vor; der Prozess ist hochoptimiert.

Flexibilität, also die Möglichkeit, sehr schnell neue Wertschöpfung aus bestehenden Strukturen zu erreichen, steht in engem Zusammenhang mit beschleunigter Umsetzung von Anforderungen und kürzeren Release-Zyklen:

ID	Wertebereich für „Flexibilität"
1	Kaum spürbare Veränderung
2	Synergieeffekte sind spürbar, Projektlaufzeiten sind etwas verringert
3	Projekte sind deutlich entlastet und werden schneller abgewickelt.
5	Projekte sind sehr stark entlastet, die Releasezyklen sind halbiert, die Budgets um 30 % gesenkt, neue Möglichkeiten werden eröffnet. Viele Services werden wiederverwendet.
8	Die Stabilität der Infrastruktur erlaubt eine starke Konzentration auf die fachliche Problematik innerhalb eines Projekts. Services werden in hohem Maße wiederverwendet, was zu hohen Einsparungen führt. Die hohe Flexibilität wird auch im Management durch schnelle Reaktionszeiten und Umsetzung von Änderungswünschen deutlich

Das folgende Beispiel illustriert die Aufarbeitung von geplanten Maßnahmen. Das Ziel in dieser Phase ist eine mehr oder weniger subjektive Beurteilung des SOA-Vorhabens, diese wird entsprechend des unternehmerischen Umfelds sehr individuell ausfallen. Die Argumentation wird so fundiert und ist die Grundlage für die darauf folgende Quantifizierung. Es folgt ein Beispiel für die Maßnahme „Einführung eines ESB" – Hintergrund ist eine Umgebung, die schon länger mit Messaging-Systemen arbeitet:

ID	Einführung eines Enterprise Service Bus	vor	nach
Details	Das Prinzip eines Enterprise Service Bus (ESB) vereinfacht die Kommunikation zwischen Diensten und Anwendungen. Ein zusätzlicher Layer mit Transformationen, Routing, Filterung und technischen Diensten erhöht die Abstraktion von Service-Beschreibungen und der Verwendung derselben. Aspekte wie Quality of Service (QoS), Security, Logging, Monitoring usw. werden vereinheitlicht. Standard-Adaptoren an Fremdsysteme erhöhen den Grad an Wiederverwendbarkeit und beschleunigen die Entwicklung.		
Kosten	Projektkosten: Der Entwicklungsaufwand reduziert sich durch die Nutzung der ESB-Funktionalität und der angebotenen technischen Dienste. Produktkosten (nicht eingerechnet): Die Anschaffung des ESB-Produkts XY beläuft sich auf Lizenzkosten von EUR 50.000,– jährlich, hinzu kommt ein Support-Vertrag von EUR 5.000,– jährlich. Die ausgiebige Nutzung erfordert verstärkte Ausbildung für Konfiguration und Integration weiterer Teilsysteme.	100	85
Vorteile	Die Nutzung von Mehrwertdiensten wie Datentransformation, inhaltsbasiertes Routing von Nachrichten oder auch Protokoll-Adaptoren entlasten die Projektkosten. Eine Infrastruktur für die Orchestrierung von Services (mittels BPEL Engine) erlaubt das flexible Umsetzen von Geschäftsprozessen und bildet die Basis für ein qualifiziertes Business Process Management. Ein durchgängiges Business Activity Monitoring wird mithilfe standardisierter Messpunkte möglich. Eine einheitliche Transportschicht mit einheitlichen Integrationspunkten (Gateways zu synchronen und asynchronen Protokollen) reduziert den Aufwand für Umsetzung und Betrieb. Das nachrichtenbasierte Bus-System lässt sich weit besser skalieren als monolithische Anwendungen. So lassen sich ggf. ältere im Batch-Modus betriebene Prozesse in Real-Time-Prozesse umwandeln.	2	8
Flexibilität	An neuen Möglichkeiten ergeben sich: ■ Der ESB ermöglicht das Beobachten der Prozesse auf Basis diverser standardisierter Messpunkte. Auf dieser Basis können SLAs, Performance und Stabilität überprüft werden. Das Business Activity Monitoring ermöglicht einen Gesamtüberblick über alle beteiligten Geschäftsprozesse, ihren Nutzungsgrad, ihre Performance und Fehlerzustände. ■ Eine Erweiterung der Prozesse um Ereignisverarbeitung (Complex Event Processing) erlaubt nun sehr einfach eine Analyse der Datenflüsse und die Einführung komplexer Geschäftsregeln zur Laufzeit.	2	5
Risiko	Die Einführung eines ESB bringt an sich kein Risiko mit sich. Die Prinzipien der Middleware, die als typische Grundlage einer SOA dienen, sind erprobt und die entsprechenden Produkte haben ihre „Kinderkrankheiten“ bereits hinter sich gelassen. Es ist allerdings darauf zu achten, dass die Verwendung des ESB im Rahmen einer funktionierenden SOA Governance erfolgt, damit der erhoffte Vorteil der höheren Agilität tatsächlich eintrifft. Auch auf Basis von ESBs lassen sich Applikationssilos entwickeln. Die Wahrscheinlichkeit dafür ist allerdings mit einem ESB deutlich herabgesetzt	3	1

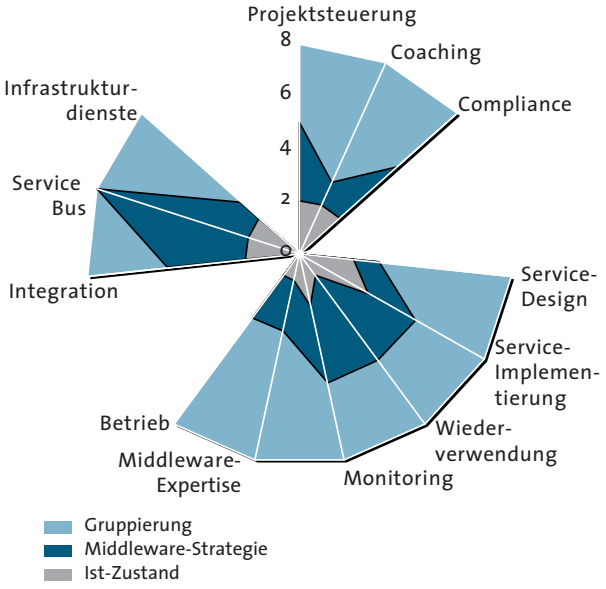


Abbildung 9: Vorteile einer ausgebauten SOA

Die Bewertungen beziehen sich auf die Situation vor und nach Umsetzung der genannten Maßnahme. Da in diesem Beispiel bereits Ansätze einer SOA in Form von Messaging und datenbasiertem Routing verwirklicht waren, wird die Ist-Situation in Bezug auf Vorteile und Flexibilität nicht mit „1“ bewertet, was eigentlich anhand des Bewertungsmaßstabs erwartet würde. Aus der Summe aller Bewertungen ergeben sich Bilder wie das Folgende, deren Wert insbesondere im Verdeutlichen sehr starker Veränderungen liegt.

Phase IV – Quantifizierung

Die Einsparungen für Entwicklungsprojekte werden auf Basis der in der zweiten und dritten Phase entworfenen Modelle und der so gewonnenen Erkenntnisse berechnet. Die erwarteten Projektkosten sowie die Investitionen, die für die SOA-Einführung geplant wurden, werden nun als Kostenkurve über die Zeitkurve gelegt und sodann den Kosten ohne SOA gegenübergestellt. Daraus ergibt sich der Break-Even-Zeitpunkt des gesamten Vorhabens.

Zunächst wird die Kostenstruktur eines Projekts (siehe Tabelle aus Phase II) zu einer Matrix erweitert, in der die in Phase III definierten und bewerteten Maßnahmen mit den verschiedenen Aspekten eines Entwicklungsprojekts in Beziehung gesetzt werden. Bei der Erstellung dieser Matrix wird schnell deutlich, ob die Erwartungen realistisch sind und an welchen Stellen sich die Vorteile verschiedener Aspekte überschneiden. Ein erfahrenes Team aus Architekten und Kennern der Unternehmensprozesse kann hierbei recht schnell zu einem gemeinsamen Verständnis von realistischen und unrealistischen Erwartungen kommen. Es erfolgt eine Nachbereitung der Ergebnisse aus Phase III. Aufgrund des besseren gemeinsamen Verständnisses der Beteiligten werden entsprechende Korrekturen vorgenommen.

Aus dem obigen fiktiven Beispiel ergibt sich eine erwartete Einsparung von 28 % bei einem durchschnittlichen Entwicklungsprojekt. Nun setzt diese Zahl voraus, dass die Bedingungen einer SOA im Endausbau vorherrschen. Zum Beispiel kann „Wiederverwendung“ nur funktionieren, wenn bereits eine kritische Masse an Services zur Wiederverwendung bereitsteht. Ein effizienter Betrieb ist erst nach einer Konsolidierung von Produkten, Anwendungen, Daten und Prozessen möglich. Wie eingangs schon vorgeschlagen, sollte sich das „Vorhaben SOA“ auf einen Zeitraum von drei bis fünf Jahren beziehen. Auf eine Zeitleiste aufgetragen wird also erst nach einer gewissen Zeit das volle Einsparpotenzial deutlich.

Erwartete Einsparungen in einem durchschnittlichen Projekt in Prozent	Vereinheitlichung der Infrastruktur		Sicherstellung von Konsistenz im Design	Vermeiden von Mehrfachentwicklung	Durchgängiges Business Activity Monitoring	Überprüfen von Standards und Richtlinien	Einsparungen summiert	Ausgleich für Mehrfachentwicklung	Einsparungen gesamt	Anteil an gesamten Projektkosten	Einsparungen gewichtet nach Ausgleich
Fronted-Anwendungen neu entwickeln		...		1			2	0	2	5	0,1
Fronted-Anwendungen verändern		...					1	0	1	5	0,1
Ausbildung Service-Verwendung	10	...					20	30	14	1	0,1
Infrastruktur-Software (Lizenzen 1 Jahr)	30	...		2			32	0	32	6	1,9
Software-Wartungsverträge (1 Jahr)	30	...		2			32	0	32	3	1
Ausbildung Service-Infrastrukturaufbau	30	...					40	30	28	1	0,3
Ausbildung Service-Infrastrukturaufbau	20	...		2			27	0	27	8	2,2
Integration und Adaption	20	...					33	0	33	10	3,3
System Management	30	...			30		63	50	32	3	0,9
Hardware (Anschaffung)	25	...		2			7	0	7	5	0,4
Betreib 1. Jahr (Hardware, Netze, Admin)	5	...		2			17	0	17	4	0,7
Externe Beratung: Technik	15	...					40	0	40	2	0,8
Center of Excellence, Governance	20	...					0	0	0	0	0
Neue Anwendungen entwickeln		...		20			35	20	28	15	4,2
Änderungen und Erweiterungen		...		5			38	20	30	14	4,3
Anwendungen mit Schnittstellen kapseln		...					45	0	45	5	2,3
Ausbildung Softwareentwicklung		...					10	0	10	1	0,1

Erwartete Einsparungen in einem durchschnittlichen Projekt in Prozent	Vereinheitlichung der Infrastruktur		Sicherstellung von Konsistenz im Design	Vermeiden von Mehrfachentwicklung	Durchgängiges Business Activity Monitoring	Überprüfen von Standards und Richtlinien	Einsparungen summiert	Ausgleich für Mehrfachentwicklung	Einsparungen gesamt	Anteil an gesamten Projektkosten	Einsparungen gewichtet nach Ausgleich
Externe Beratung: Methoden		...					20	0	20	1	0,2
Fehlerbehebung	5	...	20	5	5	3	46	40	28	4	1,1
Security	20	...				5	70	20	56	3	1,7
Logging, Auditing	20	...			5	5	75	20	60	2	1,2
Filterung, Routing, Transformation	20	...					65	20	52	2	1
Prozent Einsparungen Projektkosten		...								100	28

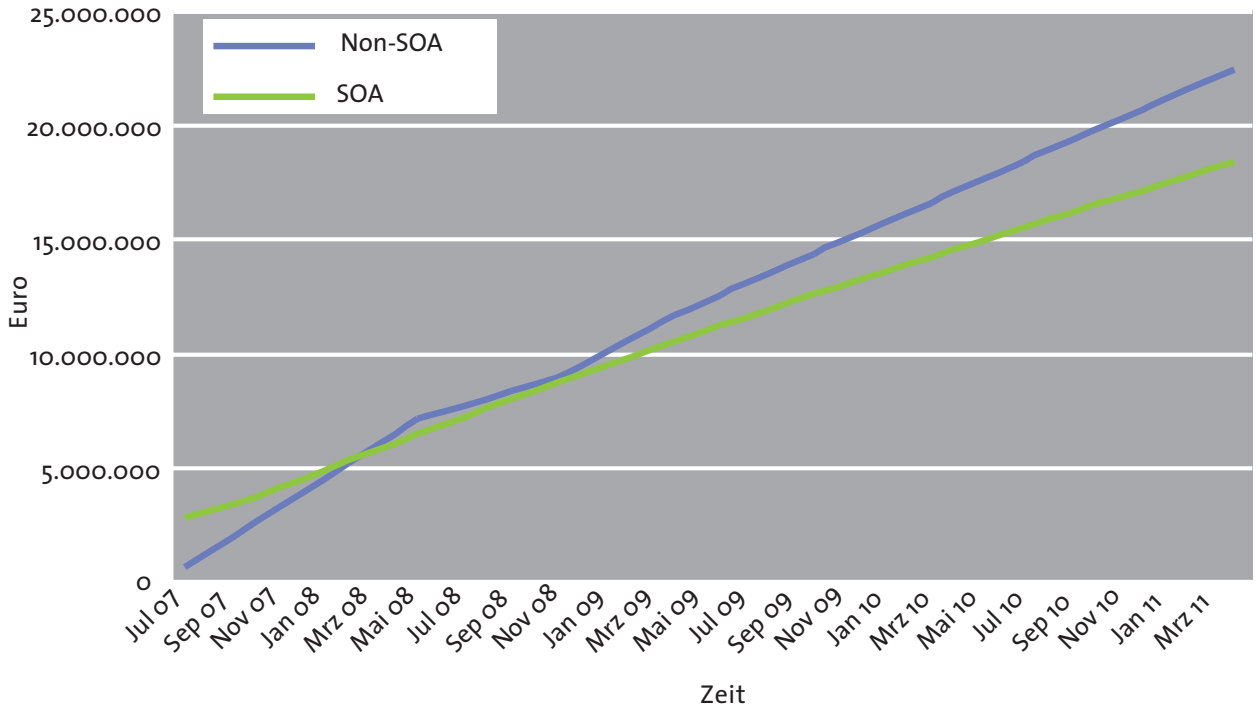


Abbildung 10: Break-Even, kumulative Kosten

Einführung der SOA-Prinzipien

Das Problem vieler strategischer Ziele ist der lange Weg, der bis zu ihrer Erreichung gegangen werden muss und zudem mit Unsicherheit gepflastert ist. Daher ist es ratsam, für die initialen Schritte Geschäftsprojekte mit geringer strategischer Bedeutung auszuwählen, die aber komplex genug sind, um das Funktionieren einer SOA beweisen zu können. Hierfür eignen sich besonders Geschäftsprozesse, die einen Endbenutzer einbinden, z. B. typischerweise über ein Self-Service-Portal. An diesen Pilotprojekten kann man Erfahrungen sammeln und die Mitarbeiter entsprechend ausbilden. Die Anforderungen solcher Projekte sollten also überschaubar sein und keine großen Abhängigkeiten zu anderen Anwendungssystemen haben, sodass Risiken, die durch mangelhafte Kommunikation entstehen, minimiert werden. Zu Beginn sollten klare numerische Ziele für die Messung des Erfolgs oder Misserfolgs der Pilotprojekte festgelegt werden.

Die Orientierung an einem SOA-Reifegrad-Modell (Maturity Model) hilft bei der langfristigen (mehrjährigen) Planung. Eine SOA-Umgebung muss inkrementell wachsen und möglichst von zwei Seiten flankiert werden, nämlich von der geschäftlichen Seite durch das Erarbeiten einer Unternehmensarchitektur und von der technischen Seite durch die Einführung von entsprechenden Middleware-Produkten sowie der Vorgabe technischer Richtlinien. Formalisierte Prozesse bei der Auswahl und dem Design neuer Services erhöht die Chance auf deren Wiederverwendung in unterschiedlichen Geschäftsbereichen. Eine strukturierte Herangehensweise, die eine Priorisierung von Projekten vornimmt und den Nutzen der entsprechenden Geschäftsprozesse evaluiert, reduziert das Risiko von Fehlentwicklungen und lenkt den Fokus auf das Wesentliche (Portfolio Management). Hierzu sind Mechanismen nötig, mit denen Service-Beschreibungen und Abhängigkeiten verwaltet werden. Dies ist ein essenzieller Aspekt bei der Verfolgung einer SOA-Strategie, der Service-Level-Agreements (SLA) und Quality-of-Service-Metriken erst möglich macht.

Service-orientierte Architekturen verfolgen als wesentliches Ziel die hohe Flexibilität und Bewegungsfreiheit im Umgang mit Geschäftsprozessen. Diese Flexibilität wird am besten durch Etablierung agiler Prozesse und Mechanismen erreicht. Dazu gehören eine sorgfältige Anforderungsanalyse im Rahmen eines iterativen Vorgehens, automatisierte Tests zur Absicherung häufiger und kurzfristiger Änderungen sowie eine effiziente Verwaltung von Abhängigkeiten zwischen Services mit Hilfe eines Service-Repository. Die Investitionen in diese Neuerungen werden sich erst lohnen, wenn die Kernziele Flexibilität, Wiederverwendung und Integrationsfähigkeit erreicht sind.

Eine ROI-Analyse wird meist zur Rechtfertigung eines Vorhabens durchgeführt. Um diese Analyse glaubwürdig zu gestalten, müssen in Bezug auf Entwicklungsprojekte und den Betrieb von Anwendungen entsprechende Metriken im Unternehmen eingeführt werden, die einen Nachweis über den Erfolg der gewählten Strategie erbringen. Die retrospektive Beurteilung von Projekten und die offene Kommunikation von Ergebnissen gehört dann zur Unternehmenskultur. Gleichzeitig darf aber der Einsatz von Metriken nicht zu einer eingeschränkten Beurteilung von Projekten führen, die als relevante Erfolgskriterien nur die Einhaltung des zeitlichen und wirtschaftlichen Projektplans berücksichtigt. Der Return on Investment liegt bei solch strategischen Veränderungen wie der Einführung einer Service-orientierten Architektur eben nur bedingt in finanziellen Ergebnissen und eher in langfristigen Vorteilen für die Handlungsfähigkeit des Unternehmens.

3.4 Migrationsplanung

Technologisch ist es ohne größere Probleme möglich, einzelne Anwendungen zu zerlegen und auf eine dienstorientierte Plattform zu migrieren. Der Nutzen für ein Unternehmen ist jedoch eher gering, wenn die Migration einzelner Anwendungen unkoordiniert erfolgt.

Der Umbau von bestehenden IT-Landschaften in Richtung einer Service-orientierten Architektur ist wie bereits erläutert immer ein längerfristiges strategisches Programm, welches von der Geschäftsführung unterstützt werden muss und sich in Abhängigkeit von der

Organisationsgröße und der bereits bestehenden Anwendungskomplexität über mehrere Jahre erstrecken kann. Die technologischen Herausforderungen einer derartigen Migration auf IT-Ebene, d. h. der Ausrichtung von Software und Infrastruktur gemäß der SOA-Prinzipen, sind jedoch beherrschbar und können teilweise bereits durch verfügbare Werkzeuge unterstützt werden. Entscheidend für den nachhaltigen Erfolg der Migration ist jedoch der Umbau der Organisation und die Definition eines SOA-Governance-Prozesses, wie im Folgenden aufgezeigt werden wird.

Drei wichtige inhaltliche Herausforderungen bei der IT- und Organisations-Transformation im Rahmen eines Migrationsprogramms sind:

- Die Definition eines SOA-Governance-Prozesses: Dieser Prozess stellt den Rahmen, in dem die Ziele, Zuständigkeiten, Rollen und Prozesse bzgl. der Weiterentwicklung der Geschäfts- und IT-Modelle sowie des Lebenszyklus der Dienste (Erstellung, Betrieb, Weiterentwicklung) in der Organisation verwaltet werden.
- Eine sinnvolle Verknüpfung der Geschäftsmodelle und der IT-Modelle, d. h. letztlich die Abbildung geschäftlicher Aktivitäten auf IT-technisch realisierte Dienste
- Kontinuierliche Qualifizierungsmaßnahmen für die Mitarbeiter

Modelle zur Beschreibung des Ist- und Sollzustandes im Unternehmen

Vor einer Migrationsplanung müssen der Istzustand sowie der Sollzustand nach der Migration definiert werden. Dies muss sowohl auf der Ebene der Geschäftsprozesse und -aktivitäten als auch auf der Ebene der IT-Fachverfahren erfolgen: Die auf Management- und Fachabteilungsebene existierenden *Organisations-, Prozess- und Geschäftsobjekt-Modelle* sind im Rahmen der Migration auf die IT, d. h. beispielsweise auf *IT-Architektur- und Betriebs-Modelle* abzubilden. Auf dieser Grundlage wird der angestrebte Zielzustand nach der Migration definiert.

Im Mittelpunkt dieser „Verknüpfung" stehen hierbei technologieunabhängige Modelle:

- Logische Referenzarchitektur, die einen Überblick über die zur Verfügung stehenden Dienste gibt
- Semantische Beschreibung der zur Verfügung stehenden Services
- Service-Level-Agreements (SLAs)
- Abbildungs- und Serviceverwaltungsmethoden

Diese Modelle bilden die logische Verknüpfung von IT und Business und müssen entsprechend den Anforderungen beider Seiten regelmäßig weiterentwickelt werden, um die propagierte Flexibilität des Service-orientierten Ansatzes zu erreichen. Eine enge und insbesondere zu Beginn der SOA-Migration aufwendige Abstimmung ist notwendig, um ein gemeinsames Gesamtmodell zu erstellen und die zwingend erforderliche Akzeptanz dieses Modells durch alle beteiligten Gruppen im Unternehmen sicherzustellen. Unabhängig von der Quelle oder Ursache des Änderungsbedarfs wirken sich neue Anforderungen mitunter erheblich auf die jeweils andere Seite aus – Geschäftsmodelle bzw. IT-Modelle. Beispielsweise kann sich eine neue gesetzliche Anforderung auf Geschäftsmodellebene über entsprechend veränderte SLAs bis auf die IT-Infrastruktur auswirken. Den Organisationseinheiten und beteiligten Personen muss von Beginn einer Migration an vermittelt werden, dass die Veränderung des Gesamtmodells zukünftig erwünscht ist und gefördert wird, da sie für den dauerhaften Erfolg des Unternehmens notwendig ist.

Identifikation von Diensten im Rahmen der SOA-Migration

Um das Geschäftsmodell auf das IT-Modell abzubilden, sind verschiedenen Herangehensweisen für das Migrations-Vorhaben möglich:

Top-down: Die Ableitung der technischen Dienste bzw. Services aus fachlichen Anforderungen wird durch eine zunehmende Verfeinerung bei dem Geschäftsmodell beginnend vorangetrieben. Der Schwerpunkt liegt hier auf einem frühzeitig abgestimmten und einheitlichen

Geschäftsmodell. Dieses Vorgehen ist zu Beginn des SOA-Aufbaus geeignet, um die Ableitung und Abstimmung eines gemeinsamen Geschäftsobjekt-Modells voranzutreiben. Für große Unternehmen mit einer Vielzahl verschiedener Geschäftbereiche birgt der Ansatz die Gefahr, viel Zeit bei der Abstimmung eines gemeinsamen Geschäftsmodells aufzuwenden.

Buttom-up: Die Ableitung von Serviceangeboten aus den Möglichkeiten, die die bestehende IT-Infrastruktur bietet, steht im Mittelpunkt. Dieser Ansatz wird vor allem von den IT-Abteilungen vertreten: Im Rahmen veränderter Anforderungen werden auf Grundlage der existierenden Anwendungslandschaft neue Services erstellt oder bestehende erweitert.

Dieses Vorgehen ist bei großen Unternehmen verbreitet, in denen die IT-Infrastruktur konsolidiert werden soll.

Mischverfahren: Eine sehr oft angewandte Methode ist das Mischverfahren: Die in den Unternehmen bereits existierenden IT- und Geschäftsmodelle werden schrittweise aufeinander abgestimmt, dafür werden sie um SOA-spezifische Aspekte ergänzt.

Ziel aller skizzierten Vorgehensmodelle ist die möglichst schnelle Erstellung eines Gesamtmodells. Nach heutigem Wissenstand ist der Umbau in Richtung SOA in nur einem Projekt nicht sinnvoll. Gründe hierfür sind die Gesamtkomplexität aufgrund der Abhängigkeiten zwischen den Teil-Modellen sowie der nicht zu unterschätzende Bedarf an Abstimmung und organisatorischen Änderungen in dem gesamten Unternehmen.

Das in der IT bereits länger bekannte und erprobte Vorgehen einer *inkrementellen und iterativen Entwicklung* scheint auch hier geeignet zu sein und lässt sich gut auf die Modell(weiter)entwicklung übertragen. Die Beauftragung und Umsetzung von Diensten muss zentral gesteuert und überwacht werden (Aufgabe der IT-Governance). Die Implementierung kann gegebenenfalls auch durch eine zentrale Stelle (Abteilung) erfolgen. Die Identifikation von Services wird zunehmend dezentral und parallel durch die jeweiligen Fachabteilungen getrieben – ein weiteres Argument für ein inkrementelles und iteratives Vorgehen bei Entwicklung und Pflege von Services.

Um beispielsweise Release-Wechsel von Diensten zu synchronisieren, ist eine iterative Bedarfssammlung mit anschießender Konsolidierung und Harmonisierung über ein festgelegtes Zeit-Intervall ein zielführendes Vorgehen. Der Aufbau des Service-Portfolios ist ein inkrementeller Vorgang.

Aufbau von Know-how für Mitarbeiter

Im Rahmen der Mirgationsplanung sind neben der Entscheidung für ein Vorgehensmodell der notwendige Wissenstransfer und die Schulung von Mitarbeitern entscheidende Faktoren. Alle mit der Erstellung und Pflege von Services befassten Mitarbeiten müssen über ein ausgeprägtes Verständnis für das Konzept SOA verfügen. Der temporäre Einsatz von externen Spezialisten ist ratsam, ersetzt aber nicht den Know-how-Aufbau im Unternehmen.

Teams, die speziell für die Pflege und Dokumentation der Modelle verantwortlich sind, müssen kontinuierlich umfangreiche Kenntnisse der jeweils relevanten Modellierungmethoden aufbauen. Die Fähigkeit, die Auswirkungen von Änderungen an einzelnen Teilmodellen auf das gemeinsame Gesamtmodell erkennen zu können, ist vor dem Hintergrund einer kontinuierlichen Weiterentwicklung bzw. Fortschreibung einer SOA essenziell. Nur so kann der Abstimmungsaufwand begrenzt werden. Fehlentwicklungen auf allen Modell-Ebenen kann so frühzeitig entgegengewirkt werden.

SOA-Migrationsplanung ist angepasste IT-Programmplanung

Die große Anzahl beteiligter Personen mit unterschiedlichen Vorkenntnissen zum Thema SOA, der damit verbundene hohe Kommunikationsaufwand sowie die hohe Komplexität des Vorhabens erfordern Managementmethoden, wie sie in großen IT-Projekten angewandt

werden. Folgende Besonderheiten des Programmmanagements sind für eine SOA-Migration nicht zu unterschätzen:

Ressourcenplanung: Aufgrund der engen Verzahnung von fachlichen Aspekten und IT-Themen, insbesondere bei der Serviceentwicklung, muss die Abhängigkeit des Projektfortschritts von Ressourcen aus den Fachabteilungen unbedingt beachtet werden.

Reporting: Regelmäßige Berichterstattung über den Fortschritt ist zwingend notwendig, insbesondere vor dem Hintergrund des zu erstellenden gemeinsamen Gesamtmodells. Im Rahmen eines iterativ inkrementellen Vorgehens und stark voneinander abhängigen Meilensteinen bezüglich der Modellentwicklung und Pflege sind formal standardisierte Reporting-Verfahren mit ausreichend personellen Ressourcen einzuplanen.

Changemanagement: Die Umsetzung einer SOA bringt große Veränderungen mit sich. So kann beispielsweise die Einführung von Prozessen zur Entwicklung und Pflege von Services auf Grundlage der Bedürfnisse der Fachabteilungen Widerstände sowohl auf IT- als auch auf Fachseite verursachen. Auf Basis einer genauen Analyse der notwendigen Veränderungen (Was und Wer?) kann eine geeignete, zielgruppenorientierte Kommunikationsplanung (Wie?) erfolgen. Der Ressourcenaufwand für das Changemanagement im Rahmen einer SOA-Migration variiert in Abhängigkeit von dem gewählten Vorgehensmodell und dem Umfang der betroffenen Unternehmensbereiche.

3.5 Value Chain – Wertschöpfungsketten

Auswirkung von SOA auf die Value Chain/ Collaborative Business

Dieses Kapitel beschreibt weitgehend aus Anwendungssicht, welche Auswirkungen und welchen Nutzen Service-orientierte Architekturen auf Wertschöpfungsketten haben; dabei wird die Zusammenarbeit von Firmen (Collaborative Business) zentral betrachtet. Zur Erläuterung der Anforderungen und Auswirkungen aus geschäftlicher Sicht werden Beispiele aus dem Umfeld des Lieferketten-Managements (Supply Chain Management) und der Logistik herangezogen, weil in diesen Bereichen bereits jetzt hohe Anforderungen an die Struktur und Leistungsfähigkeit der Integration gestellt werden. Darüber hinaus sind diese Bereiche geschäftlich ausreichend komplex und strukturiert, um als Anschauungsobjekt zu dienen. Natürlich beschränkt sich eine SOA auch bei externen Wertschöpfungsketten nicht auf diese Branchen.

Grenzen und Möglichkeiten der Weiterentwicklung bzw. Erschließung von Geschäftsbereichen mithilfe von SOA werden in diesem Kapitel angedacht.

Collaborative Business und Value Chain

Wertschöpfungsketten existieren inner- und außerhalb von Firmen. Eine externe Wertschöpfungskette geht dabei deutlich über das einfache Lieferanten-/Konsumenten-Konzept hinaus. Die Abgrenzung zwischen einfachen Lieferanten-Beziehungen und externen Wertschöpfungsketten ist fließend. Hier kann sie eher anhand von Beispielen als durch formale Definitionen dargestellt werden. Die Begriffe (externe) Wertschöpfungskette und Collaborative Business/Value Chain werden im Folgenden synonym verwendet.

Wichtige Charakteristika externer Wertschöpfungsketten sind:

- Die Zusammenarbeit zwischen den Geschäftspartnern ist komplex, eng aufeinander abgestimmt und geht weit über einfache Lieferanten-/Konsumenten-Beziehung hinaus.
- Essentielle Wertschöpfung findet auch zwischen den Beteiligten statt (Mehrwertleistungen).
- Die Struktur der Zusammenarbeit ist oft komplex mit längeren, wiederholten Abfolgen von logistischen und datentechnischen Aktivitäten der Lieferanten und Abnehmern – daher auch „Kette“; die Struktur kann auch komplex verzweigt sein, daher auch oft der Begriff „Wertschöpfungsnetzwerk“.

- Der Organisations- und Automatisierungsgrad ist hoch; komplexe fach- und branchenspezifische Prozess- und Datenmuster kommen zum Einsatz.
- Formale und nicht funktionale Anforderungen (etwa zeitliche Kopplung) sind hoch.

In der Praxis gibt es eine Reihe von Beispielen:

- SCM – Supply Chain Management:
  In diesem Umfeld, häufig in der Automobilbranche, arbeiten Hersteller in enger Abstimmung mit Zulieferern zusammen. Die Zulieferer werden je nach ihrer Position in der Wertschöpfungskette als 1st-, 2nd oder 3rd-Tier-Supplier bezeichnet. Im Rahmen des SCM werden Produktionsabläufe zwischen Herstellern und Zulieferern verteilt und abgestimmt. Häufig angewandte Modelle sind die Lieferung Just-In-Time und Just-In-Sequence, bei denen die Fertigungskomponenten genau zum richtigen Zeitpunkt und in der passenden Reihenfolge der Montage durch die Zulieferer bereitgestellt werden müssen. Dieses Vorgehen setzt eine sehr intensive Abstimmung und leistungsfähige IT-Integration über die Grenzen der einzelnen Beteiligten voraus. Daher ist SCM häufig ein Vorbild und Benchmark für den Integrationsstatus im Collaborative Business. Wesentliche Wertschöpfungsbeiträge von SCM bestehen in der Reduktion von Lagerbeständen entlang der Fertigungskette, einer deutlich höheren Wertigkeit der Komponenten gegenüber dem Produkthersteller und damit einer größeren Spezialisierung und Effizienz der Komponentenfertiger.
- CPFR – Collaborative Planning, Fulfilment and Replenishment:
  Diese integrierte Nachschubsteuerung erfolgt vorwiegend im Handel. Ideen und Anforderungen der Integration sind dabei ähnlich wie im SCM. Die zeitlichen und strukturellen Anforderungen fallen in diesem Kontext allerdings deutlich geringer aus (so entfallen komplexe Konstruktionsinformationen und Stücklisten). Einschränkungen bezüglich der Lagerdauer und – speziell bei Lebensmitteln – besondere Transportarten steigern die Komplexität. Wesentliche Wertschöpfungsbeiträge von CPFR sind die Reduktion von Sicherheits-Lagerbeständen sowie eine effektivere Transportlogistik durch einen größeren Planungshorizont.
- In der Telekommunikation spielt eine starke Zusammenarbeit zwischen Service-Lieferanten (gegenüber Kunden), Technologie-/Infrastruktur-Anbietern (etwa Netze) und spezialisierten Service-Dienstleistern (etwa Abrechnung) eine zunehmend wichtige Rolle. Die konzeptionelle Strukturierung liegt hier noch hinter SCM und CPFR zurück; der wesentliche Wertschöpfungsbeitrag besteht hier in der Spezialisierung und somit verstärkten Effizienz der einzelnen Beteiligten.

Die Strukturen im Collaborative Business, also die an Geschäftsprozessen orientierten Beziehungen zwischen den einzelnen Geschäftspartnern, sind im Laufe der Zeit immer komplexer geworden. Sie lassen sich folgendermaßen untergliedern:

- Bilaterale Kommunikation (Einzel-Schnittstellen):
  Auch heute kommunizieren Unternehmen untereinander überwiegend auf diese Art. Die bilaterale Kommunikation bildet zugleich auch die Grundlage für alle nachfolgenden Kommunikationsarten. Die klassische Schnittstelle hat eine stark reduzierte Öffentlichkeit – nämlich nur die unmittelbar Beteiligten – und basiert auf bilateralen Vereinbarungen (Syntax, Semantik, Service-Levels).
- Kommunikation in Ketten:
  Bekanntestes Beispiel hierfür sind Lieferketten (SCM). Dort wird die gesamte Fertigung eines Produkts hierarchisch in Komponenten gegliedert. Die Hersteller einzelner Komponenten sind dabei jeweils auch Konsumenten von Teilkomponenten oder Rohstoffen vorgelagerter Hersteller. Oft entstehen so „Ketten“, bei denen eine 1:n-Bezieung zwischen einem Komponenten-Fertiger und seinen Zulieferern besteht (n-Tier-Supplier-Modell). Durch die 1:n-Beziehung entstehen häufig Vereinbarungen einer breiteren Öffentlichkeit, die zu einheitlichen Beschreibungen, typisch innerhalb einer Branche, geführt haben (Beispiel: VDA-Normen für Lieferabrufmeldungen in der Automobilbranche). Service-Levels werden üblicherweise durch den jeweiligen Abnehmer vorgegeben und individualvertraglich mit den n Zulieferern vereinbart.

- Kommunikation in Netzen:
  Hierbei handelt es sich um die Erweiterung bzw. Multiplikation des Ketten-Modells. Ein 1st-Tier-Supplier ist gleichzeitig für mehrere Hersteller tätig. Dies ist heute das Leading-Edge-Paradigma, aktuell auf eng abgestimmte und statisch geplante Geschäftsprozesse ausgelegt. Ähnlich wie bei den Ketten sind diese Netze allenfalls innerhalb der Branche öffentlich und werden häufig organisatorisch von den Endproduzenten dominiert. Die geforderte Flexibilität und Anpassung wird jeweils entsprechend der Markstellung „nach unten durchgedrückt". Die Flexibilität, die mehr Effizienz in dem gesamten Netz liefern würde, scheitert häufig an der mangelnden Flexibilität einzelner Beteiligter, meistens aufgrund sehr unflexibler Binnen-IT-Strukturen.
- Dynamische Netze:
  Dies ist die Weiterführung des Netzwerkmodells durch permanente kurzfristige Veränderungen (dynamisches Supplier-Netz). Ein einmal etabliertes Netz bleibt nicht statisch, sondern ändert sich dynamisch. Dies kann unterschiedliche Auslöser haben: Die Entwicklung von neuen Produkten und Gesamt-Services erfordert zunehmend schnelle Anpassungen oder Neubildung von Netzwerkteilen. Beispiele solcher schnelleren Produktentstehungszyklen gibt es derzeit im Telekommunikationsbereich. Ein anderer Auslöser kann das selbst optimierende Wertschöpfungsnetz sein, bei dem dynamisch fortlaufend Anpassungen an neue Service-/Leistungsanbieter vorgenommen werden.

Klassisches Collaborative Business

Nach dem Überblick, welche Aufgaben und Fragestellungen mit Wertschöpfungsketten und dem Collaborative Business verbunden sind, sollen diese nachfolgend aus der klassischen IT-Sicht dargestellt werden. Dabei werden auch die Veränderungen, die SOA mit sich bringt, aus der Perspektive der bestehenden IT-Integration beleuchtet– auch wenn SOA kein (reines) IT-Thema ist und sehr starken Bezug zu den Prozessen und der Organisation hat, bewirkt SOA natürlich eben auch Änderungen in der IT-Welt. Die nachfolgenden Beispiele sollen verdeutlichen, wie der Einsatz von SOA die entsprechende konventionelle (IT-gestützte) Abwicklung optimieren kann.

Struktur der Kommunikation

Die klassische IT-Integration entlang der Wertschöpfungskette lässt sich allgemein wie folgt charakterisieren:

- Es gibt feste Datenformate – meistens individuelle für jede Schnittstelle.
- Die Kommunikation findet bilateral „Punkt zu Punkt" statt und bleibt hinsichtlich der Struktur über lange Zeit erhalten.
- Eine Standardisierung der Formate findet maximal auf Ebene von branchenspezifischen Austauschformaten statt.

Diese bestehenden Ansätze können schrittweise nach Ansätzen der Serviceorientierung überführt werden. Auf diesen Aspekt wird später weiter eingegangen.
Technisch gibt es zwei wichtige Arten der Kommunikation und Integration:

- Datei-Austausch
- Portal-Hubs

Der Dateiaustausch orientiert sich an klassischen Dokumenten und bildet diese oft nach. Dateien werden durch unterschiedliche Transportmedien versendet, etwa durch Mail, gemeinsam benutzte Message-Queues oder auch spezielle Datei-Übertragungsprotokolle wie ftp/tftp. Charakteristisch für diese Art der Integration ist eine Verzögerung. Oft werden auf Anwendungsebene zusätzliche Maßnahmen realisiert, um Qualitätsanforderungen bei der Übertragung sicherzustellen. Dies betrifft etwa Transaktionalität, Datenschutz und Datensicherheit.

Beispiele für einen Dateiaustausch sind Bestellungen, Lieferabrufe oder Abgleich von Stammdaten. Aus Sicht der Serviceorientierung ändern sich einige dieser Übertragungen nicht wesentlich. Eine klassische Bestellung mit einer Liste aus Bestellpositionen, Lieferkonditionen und Empfänger wird sich aus geschäftlicher Sicht kaum ändern.

Technische Vorteile aufgrund einer einfacheren Integration oder der Nutzung abgesicherter Übertragungsarten sollen hier nicht betrachtet werden. Ein Bestellservice kann dagegen, etwa durch engere zeitliche Kopplung, eine unmittelbare Rückmeldung über vorhandene Bestände und damit zugesicherte oder voraussichtliche Lieferzeitpunkte ermöglichen. Da sich die Geschäftslogik nicht notwendigerweise ändert, ist ein Umstieg in einzelnen schnellen Schritten auf die SOA-Techniken möglich. Um die Vorteile von SOA weitreichend nutzen zu können, muss mitunter jedoch die geschäftliche Aufgabenstellung modifiziert bzw. neu formuliert werden.

Eine modernere Integration erfolgt durch Portal-Hubs. Diese lassen sich folgendermaßen charakterisieren:

- oft 1:n-Beziehung (1 Einkäufer, Ausschreiber und n Lieferanten, Bieter)
- oft ist der Betreiber/Owner auch aus Prozess-Sicht ein Hauptbeteiligter
- Die technische Vorgaben sind oft proprietär oder berücksichtigen maximal Branchen-Standards (Beispiel BMI-KAT für Katalog-Daten)

Beispiele für solche Portal-Hubs sind etwa eProcurement-Bestellplattformen. Diese beinhalten im einfachsten Fall einen Austausch von Katalog-Daten (über die üblichen Wege des Datei-Transfers). Auswahl und Bestellabläufe werden dabei stark automatisiert und integriert. Für diese Aufgaben kommen intern Workflows und Regel-Interpreter für Geschäftsregeln zum Einsatz – etwa zur Steuerung von Freigabeverfahren. Gestützt durch entsprechende Rahmenverträge können Bestell-Abläufe direkt vom Nutzer ausgelöst werden. Die eigentlichen Bestellungen werden dann auf konventionellem Wege durch Dateitransfer abgewickelt.

Organisation und Herausforderungen

Bereits die konventionelle Integration muss ähnliche Anforderungen erfüllen wie die Integration über SOA. Dies ist ein Vorteil bei der schrittweisen Nutzung von SOA, weil keine grundlegend abweichenden Organisationsaufgaben zu lösen sind. Da sich die Ausrichtung von SOA grundsätzlich an den Geschäftsprozessen und deren Logik orientiert, sind die organisatorischen Anforderungen von SOA in der Wertschöpfungskette die Anforderungen des normalen Geschäftsablaufs.

Sobald die Integration die bilaterale Kommunikation verlässt und mehrere Geschäftspartner miteinander vereint, wird es notwendig, Standards bei Dateiaustausch und Übertragung zu vereinbaren. Bislang war dies Aufgabe entsprechender Fach- und IT-Branchen-Gremien, wie etwa des VDA (Verband der Deutschen Automobilindustrie), BVL (Bundesverband für Logistik) oder BME (Bundesverband für Materialwirtschaft und Einkauf).

Die bereits existierenden Organisationen erfüllen im Kontext von SOA die gleiche Aufgabe: die Definition allgemein verbindlicher Standards. Die Bedeutung der Standardisierung wird künftig weiter steigen. Darüber hinaus wird die durch SOA mögliche größere Flexibilität auch in den Standards abgebildet. Anderenfalls würde der Engpass an Flexibilität von der IT-Technik auf die fachliche Spezifikation und Definition verlagert.

Wie ändert SOA das Collaborative Business

In diesem Abschnitt werden die Veränderungen, die SOA für die Abbildung der Wertschöpfungskette mit sich bringt, dargestellt. Dabei werden die Beispiele aus den vorhergegangenen Abschnitten aufgegriffen und weiterentwickelt.

Struktur der Kommunikation

Hinsichtlich der Struktur der Kommunikation bringt SOA einige wesentliche Veränderungen und eine große Zahl von Detailverbesserungen mit sich.

Zu den wesentlichen Veränderungen zählt die grundsätzliche Offenheit der Kommunikation. Wenn SOA im Collaborative Business eingesetzt wird, werden Services öffentlich angeboten – wobei die Öffentlichkeit natürlich, je nach Einsatzfeld, auch eingeschränkt werden kann. Der

Zugang zu einem Service, anders als zu einer bilateralen Schnittstelle, erfolgt durch Vermittlung eines Verzeichnisses (bzw. Verzeichnis-Dienstes). Damit ist die Zuordnung zwischen Dienst-Suchendem und Dienst-Anbieter grundsätzlich dynamisch.

Diese Offenheit benötigt sehr ausführliche Beschreibungen, die wiederum nicht nur bilatereal gültig sein dürfen, sondern zumindest global für die Branche zu definieren sind. Diese Beschreibungen beinhalten nicht nur Schnittstellen, sondern auch Daten- und Prozessmodelle. Diese bilden im jeweiligen Verzeichnis zusammen eine formale Semantik der angebotenen Services. Allgemein in der jeweiligen Branche akzeptierte Daten- und Prozessmodelle sind daher notwendig, wenn SOA in der öffentlichen Wertschöpfungskette nicht auf einige wenige triviale Dienste beschränkt sein soll. Diese globale Prozess-Sicht und -Denkweise ist die eigentliche Herausforderung. Unterstrichen wird dies dadurch, dass in vielen Branchen standardisierte Prozessmodelle entstehen – geradezu eingefordert werden, um so ein Benchmarking und entsprechende Prozessverbesserungen herbeizuführen. Stellvertretend hierfür sei wieder der Bereich Supply Chain Management genannt, für den mit SCOR (Supply Chain Optimization Reference Model) eine solches branchenspezifisches Prozessmodell existiert. Da individuell definierte Prozesse oft ein erhebliches Hindernis für die (teileweise) Nutzung von Standard-Software (etwa ERP) darstellen, werden auch im Kontext von Standard-Software standardisierte Prozesse gefordert.

Zusätzlich zu den funktionalen Beschreibungen in den öffentlichen Service-Verzeichnissen müssen auch nicht funktionale Anforderungen (etwa Verfügbarkeiten, zugesicherte Bearbeitungszeiten) hinterlegt werden. Diese Service-Level- Agreements sollten auch in standardisierter Form dargestellt werden. Dies unterstützt wiederum die Entwicklung neuer Dienstleistungs- und Geschäftsmodelle (s. u.).

Neben der beschriebenen wesentlichen Veränderung gibt es eine Reihe von graduellen Verbesserungen, die SOA auch für die Geschäftsmodelle erbringt, die im Wesentlichen noch konventionell aufgebaut sind.

Die Granularität der externen Kommunikation ist bei SOA häufig höher – es werden also grundsätzlich mehr, dafür aber kleinere logische Datenpakete ausgetauscht (Beispiel: Eine einzelne Bestellung zeitnah anstelle einer Sammelbestellung pro Tag) Der Zuschnitt orientiert sich dabei wieder – wie bei SOA üblich – stärker an prozesslogischen Business-Objekten anstatt an von technischen Datensätzen oder Tabellen. Firmen, die SOA bereits intern nutzen, können die vorhandenen Services mit ihrer feineren Granularität auch für die externe Kommunikation nutzen. Das Maß der externen Granularität – also die Dateneinheiten, die zwischen Firmen ausgetausch werden – wird sicher gröber sein als das interne (Beispiel: Extern: Komplette Bestellung – Intern: Bestellpositionen). Die Bestimmung einer optimalen externen Granularität wird eine kontinuierliche Aufgabe beim Lösungsdesign sein, unterstützt wird diese durch die Einführung von branchenspezifischen Prozessmodellen (s. o.).

SOA steigert darüber hinaus zeitliche Kohärenz der Abläufe. So werden einzelnen Abfragen „life“ unterstützt, wo sonst eher der statische und langsame Austausch von großen Datenmengen das Zusammenspiel einengte. Diese Agilität leitet sich aus dem zunehmenden betriebsinternen Einsatz von SOA ab.

Schließlich bietet die technische Implementierung von SOA, wenn sie etwa auf der Basis von SOAP erfolgt, eine flexible und mehrfache Nutzung. Ein Service auf SOAP-Basis lässt sich üblicherweise sehr schnell sowohl für die Kommunikation zwischen Systemen als auch für die Integration in ein Anwenderportal nutzen, wodurch wiederum die Wiederverwendung steigt.

Der Einfluss dieser graduellen Verbesserungen ist nicht zu unterschätzen, denn häufig ist es die Summe einiger negativen „Kleinigkeiten“, welche die Nutzung eines Geschäftmodells scheitern lässt.

Veränderungen bestehender Strukturen des Collaborative Business

Die Veränderungen der Wertschöpfungsketten auf Basis von SOA sind unterschiedlich motiviert: firmenintern (SOA

intern) und firmenextern aufgrund der Geschäftspartner oder des Marktes (SOA extern).

Firmenintern motiviert definieren Teile der vorhandenen intern genutzten Services auch die SOA-Außenansicht des Unternehmens. Je weiter, konsequenter und klarer die Definitionen und Umsetzungen der internen SOA-Anteile sind, um so einfacher und effektiver ist die Darstellung dieser Services für die Außenwelt möglich.

Bei der Klassifikation und dem SOA-Design (dem SOA-Domänenmodell) sollte die Möglichkeit einer externen Nutzung eines Service frühzeitig berücksichtigt werden. Üblicherweise werden interne Services nicht unmittelbar für eine externe Nutzung freigegeben; hierfür ist eine entsprechende Sicherheitsstruktur mit externer Authentifizierung und Autorisierung sowie Abschottung durch Netzwerk-Sicherheitseinrichtungen (Firewalls und Bildung von „Demilitarisierten Zonen (DMZ)“) notwendig. Die Berücksichtigung eines möglichen logischen Exports eines Service unterstützt jedoch die Wiederverwendung. Die Ausrichtung der Services an branchenspezifischen Prozessmodellen – sowohl nach innen wie nach außen – vereinfacht das Vorgehen. Nach außen entsprechen die exportierten Services so den allgemein akzeptierten Prozess-Beschreibungen; nach innen werden aufwändige und Geschäftslogik-intensive Übersetzungen überflüssig.

Eine externe Motivation besteht darin, dass SOA-Anforderungen von einzelnen Geschäftspartnern oder dem Markt als Ganzem an das Unternehmen herangetragen werden. Typische Beispiele aus der Vergangenheit können hier als Vorbild dienen: So erwarteten Automobilhersteller von ihren (1st-Tier-) Zulieferern die technische Kompatibilität zur VDA-Standards (EDI). Der Ablehnung dieser gesetzten Infrastruktur folgte oft die Streichung von der Lieferantenliste. Es entsteht also in einigen Marktbereichen der Druck, SOA-Services anbieten zu müssen.

Grundsätzlich können externe Services auch dargestellt werden, ohne die Binnenstruktur dem SOA-Paradigma anzupassen. Dieser Ansatz einer SOA-Kapselung wird sehr erfolgreich bei alten Legacy-Systemen eingesetzt, deren Nutzungsdauer so verlängert wird. Im Rahmen dieses Ansatzes steigt mit zunehmender Zahl externer Services der Aufwand entsprechend. Für die SOA-Strategie ist eine frühzeitige Analyse der externen und internen Anforderungen ratsam, inklusive der damit verbundenen Rentabilitätsbetrachtung.

Im Folgenden wird diese „abstrakte“ Darstellung mit einigen Beispielen aus dem klassischen Collaborative Business und möglichen Weiterentwicklungsszenarien ergänzt.

Eine klassische auf Dateien aufsetzende Kommunikation ist grundsätzlich einfach auf das SOA-Paradigma umstellbar (etwa auf SOAP). Wie bereits erläutert hat eine aus Sicht der Geschäftslogik völlig identische Abbildung („Technischer Put“) einige Vorteile, so etwa keine veränderte Semantik von der Schnittstelle zum Service und damit reduzierte Umstellungskosten. Aufgrund der geringen Ausrichtung auf branchenspezifische Prozessmodelle findet ein solcher Service jedoch möglicherweise keine große externe Akzeptanz. Ein größerer Nachteil besteht darin, dass ein solcher Service die erweiterten Möglichkeiten oft nicht ausschöpft. Die klassische Bestellung beinhaltet eine große Anzahl von Bestellpositionen und erwartet oft keine explizite Rückmeldung. Bei einigen Lieferabrufen wird sogar vertraglich fixiert, dass Bestellungen wie gefordert geliefert werden müssen, quasi nach dem Prinzip von Befehl und Gehorsam. Die feinere Granularität aus SOA-Perspektive legt dagegen eine individuelle Abstimmung einzelner Lieferpositionen nahe. So kann auf eine Bestellung eine direkte Rückantwort bezüglich der Verfügbarkeit erfolgen – auch auf Basis einzelner Lieferpositionen, gegebenenfalls sogar mit der Aufteilung eines Lieferauftrags nach unterschiedlichen Mengen und Lieferzeiten. Solche Verfügbarkeitsabstimmungen finden heute bereits häufig im B2C-Umfeld statt (praktisch jedes große Kundenbestellportal bietet es an). Auch im B2B-Kontext gestatten aktuelle Verfügbarkeitsinformationen eine wesentlich größere Flexibilität und damit auch flexiblere Geschäftsmodelle.

Modernere Anwendungen wie Portal-Hubs können ebenfalls von SOA profitieren. So kann ein Bestell-Portal direkt über Service-Abfragen mit den Lieferanten

kommunizieren, ohne statische Bestell-Kataloge. Der Vorteil liegt auch hier in der Aktualität. So können etwa Bestellgüter, die in einem bestimmten Zeitraum für einen Kunden oder einen bestimmten Lieferort nicht verfügbar sind, als nicht bestellbar gemeldet werden. Die Lieferfähigkeit kann sich kurzfristig ändern, was durch klassische Bestellungen schwerer abzubilden ist.

Durch die feinere Granularität der externen Geschäftslogik lassen sich neue oder erweiterte Bearbeitungsabläufe viel einfacher implementieren, ohne dabei die vorhandenen Basis-Services zu modifizieren. Als Beispiel dient wieder ein moderner Bestell-Service mit zugesichertem Liefertermin und Preis: In einem Szenario kann bei einer Bestellung der Preis minimiert werden, zu Lasten des Liefertermins; in einem zweiten Szenario kann der Liefertermin vorgegeben sein, ggf. unter Anpassung des Preises. Beide Szenarien lassen sich auf Basis der gleichen exportierten Services erreichen, nur durch unterschiedliche Abfolge und Steuerung der einzelnen Service-Aufrufe; eine Veränderung der vorhandenen exportierten Services ist dabei nicht notwendig.

Diese Flexibilität muss im Einzelfall im Rahmen der technischen Möglichkeiten abgebildet werden. Die Forderung nach Performanz kann extern genauso wie intern das Design von Services beeinflussen. Oft wird bei erhöhten Performanz-Anforderungen eine starke Vergröberung der prozessfachlichen Funktionalität von Services und damit deren eingeschränkte Flexibilität vorgeschlagen. Tatsächlich bietet aber gerade eine hohe Flexibilität im Rahmen der Prozess-fachlichen Anforderungen viel Potenzial zur Optimierung: Bei der Abbildung der Prozess-fachlichen Anforderung durch Einsatz der jeweils genau passenden Services (Beispiel: ausgewählte Kopfinformationen einer Bestellung anstelle der gesamten Bestellung) und ggf. auch Prüfung des fachlichen Prozesses selbst (wozu wird diese Information hier eigentlich benötigt?). Die Möglichkeit zur technischen Optimierung ist zusätzlich gegeben; diese findet aber strikt gekapselt innerhalb der Services statt und beeinflusst nicht die Prozess-Logik.

Neue Geschäftsrollen und -Modelle

Neben der Flexibilisierung bestehender Geschäftsmodelle unterstützt SOA im Collaborative Business auch die Entstehung neuer Geschäftsmodelle, die die klassischen Erzeuger-/Verbraucher-Muster ergänzen.

- Beispiel: Value-Add Service Provider

Bei modernem Transport und im Breich Logistik bestehen heute schon Muster der Zusammenarbeit, die richtungsweisend für collaborative SOA-Ansätze sein können. Auf der Basis von bestehenden elementaren Service-Angeboten entstehen neue abgeleitete Angebote, die einen zusätzlichen Mehrwert bieten. In der Logistik haben sich Spezialisierungen wie die Konsignationslogistik oder die Verpackungs- oder Montage-Logistik entwickelt.

Im Bereich der IT-Services kann beispielsweise eine Bündelung und Veredelung von bestehenden IT-Services von Dritten erfolgen. Das Beispiel Sendungsverfolgung soll dies demonstrieren: Im Collaborative SOA stellen Logistik-Dienstleister, also Bahn, Reedereien, Luftlinien, Speditionen und weitere Prozess-Beteiligte, wie Hersteller, Abnehmer oder Zoll, definierte Service-Schnittstellen bereit. Ein SOA-Mehrwert-Dienstleister kann nun diese Informationen über diese veröffentlichten Service-Schnittstellen sammeln und kunden- und auftragsgerecht aggregieren und als Mehrwert-Service für andere Prozessbeteiligte zur Verfügung stellen. Erweiterte Services können etwa die Bildung von Historien sein, spezifische Auswertungen etwa nach Service-Qualität der Logistikleistungen oder auch vorausschauende Prozess-Überwachung.

Dieses Zusammenspiel von Informationsanbietern und -Veredelern kann auch klassisch, ohne Serviceorientierung, umgesetzt werden. Dabei müssten aber eine Vielzahl individueller Schnittstellen vereinbart und implementiert werden. Bei einer effektiven Implementierung führt dies schnell zum Einsatz einer EAI-Plattform und kommt so der technischen Implementierung eines Enterprise Service Bus (ESB), einer technischen Grundlage von SOA, nahe. Der Vorteil von SOA gegenüber einem rein technischen EAI-Ansatz liegt in der konsequenten

fachlichen Ausrichtung der Services: Die fachlichen Anforderungen initiieren und steuern die Zusammenarbeit. Zusätzlich sind Services auch fachlich standardisiert beschrieben und nicht nur technisch. Durch diese standardisierte fachliche Beschreibung und flexible technische Implementierung wird es möglich, bestehende Dienstleister-Beziehungen schnell zu verändern – beispielsweise Ersatz oder Ergänzung einer weiteren Spedition oder Airline über deren standardisierte Services. Weiter wird so die schnelle Umsetzung von Mehrwert-Dienstleistungen auf der Basis der bestehenden Services möglich – beispielsweise eines Vermittlungsdienstes für unterschiedliche Transportdienstleistungen über mehrere Dienstleistungsanbieter.

- Beispiel: Spezialdienstleistungen

Spezialisiertes Geschäftswissen lässt sich durch SOA sehr effektiv nutzen und für Abnehmer nutzbar machen. Ein Beispiel – wiederum aus dem Logistik-Umfeld – soll dies darstellen: Die Erzeugung von Frachtdokumenten ist mittlerweile eine sehr spezialisierte Aufgabe, die ein hohes Maß an Spezial-Know-how benötigt. Frachtdokumente müssen die sehr komplexen Gefahrgutverordnungen berücksichtigen, die beispielsweise den gleichzeitigen Transport bestimmter Materialien ausschließen oder besondere Transport- oder Verpackungsregeln erforderlich machen. Transportmittel (Bahn, LKW, Flugzeug), Routen und Zeiten sind ebenfalls von den Ladungsinhalten abhängig. Darüber hinaus sind die Regelungen länderspezifisch. Zoll- oder Exportbestimmungen sind ähnlich komplex.

Ein spezialisierter Service-Dienstleister kann seine Expertise anderen Untenehmen anbieten, die diese Expertise in die eigenen Geschäftsabläufe integrieren. Über geeignete öffentliche Services kann so das Wissen des Service-Anbieters genutzt werden. Die Bereitstellung des besonderen Know-hows als Service unterstützt eine flexible und effektive Umsetzung, die sehr zur Akzeptanz der Lösung beiträgt. Eine konventionelle Implementierung auf Basis von statischen Schnittstellen – etwa nach EDI – würde die sehr rasch wechselnden Anforderungen behindern. Dazu folgendes Beispielszenario: Häufig und kurzfristig gibt es Veränderungen bei juristischen Verordnungen (EU-Recht), die nicht immer die Struktur der Aus- und Eingabedaten unverändert lassen. Ensprechend schnelle Änderungen von klassischen öffentlichen Standards auf der Basis von EDI sind oft genug aufgrund des statischen technischen Designs nicht möglich.

- Beispiel: Service-Broker für öffentliche Services

Mit der Verbreitung von öffentlich angebotenen Services (Service-Makrtplatz) wird die Suche nach dem jeweils passenden Service durch die Anwender für die fachliche Anforderung eine wichtige und spezialisierte Aufgabe. Ein typisches Beispiel außerhalb von SOA im Internet sind Suchmaschinen für Webseiten. Die weitere Verbreitung von öffentlich angebotenen Services führt zu Spezialisierung und Wettbewerb zwischen den Anbietern dieser Services. Ein Service-Broker, selbst ein Service, nimmt eine Spezifikation für einen gesuchten Service entgegen und vermittelt einen passenden Service-Anbieter. In der einfachsten Version ist ein solcher Service-Broker ein normaler Verzeichnisdienst. Stehen neben den funktionalen Service-Beschreibungen auch weitere nicht funktionale Informationen zur Verfügung, kann auch die Vermittlung gezielter erfolgen.

So kann ein intelligenter Service-Broker den billigsten, den schnellsten oder den Service mit der größten Verfügbarkeit vermitteln. Ein spezialisierter Broker kann auch selbst eine höhere Verfügbarkeit anbieten, indem er mehrere, unabhängige Service-Anbieter dynamisch vermitteln kann. Die Geschäftsidee für einen solchen Vermittlungsdienst kann etwa auf Vermittlungsprovisionen der beauftragten Services beruhen oder aber auf Preismargen auf die Vermittlung des günstigsten Service (Anteil der erzielten Preisreduzierung).

- Beispiel: Service-/Software-Outsourcing: Software as a Service (SaaS)

Services, die von einem Service-Anbieter gegen eine Nutzungsgebühr zur Verfügung gestellt werden, können klassischen Software-Kauf oder Miete ersetzen. So kann etwa in der Logistik das Bestandsmanagement als ein Service

angeboten und genutzt werden. Da der Service-Anbieter seinen Software-Service für möglichst viele Anwender mit unterschiedlichen IT-Systemen bereitstellen will, ist hier eine nahtlose Integration in die betrieblichen IT-Systeme der Service-Käufer sehr wichtig.

Die Bezahlung eines solchen Service kann etwa pauschal (Flat-Rate), nach individueller Nutzung (pay-per-use) oder zusätzlich nach erzieltem Nutzen erfolgen. Bei letzterem Modell könnte etwa eine erzielte Bestandsreduktion bei gehaltenen Service-Level-Agreements.

- Beispiel: Technologie-Integratoren

Der schrittweise Übergang zwischen dem klassischen in das Service-orientierte Collaborative Business bietet eine weitere Grundlage für technische Services. Technische Übersetzer können etwa zwischen modernen Services und klassischen EDI-Formaten vermitteln. Der Vorteil liegt in der Reduzierung des Aufwands für die Prozess-Anrainer für Dienstleistungen, die entweder nur zeitlich befristet benötigt werden (Übergangslösung) oder aber aufgrund der geringen Gesamtnutzung keine Investition rechtfertigen. Solche technischen Integrations-Services sind in der Vergangenheit bekannt etwa durch den Betrieb von Web-EDI-Lösungen.

Organisation, Herausforderungen

Abschließend sollen hier die kritischen Erfolgsfaktoren und allgemeinen Herausforderungen für den Einsatz von SOA im Collaborative Business zusammen gestellt werden.

Eine wichtige Bedeutung auch vom öffentlichen SOA-Einsatz kommt den Verzeichnisdiensten und öffentlichen Repositories zu. Die Wichtigkeit ist ähnlich wie beim internen Einsatz, da die Repositories aus einer reinen Sammlung von einzelnen Services nebeneinander, ein sinnvolles und wertsteigerndes Miteinander machen. Services, die ein Anbieter einfach nur öffentlich zugänglich macht, führen letztlich wieder zu statischen bilateralen Geschäftsbeziehungen, die möglicherweise nur auf eine neue technische Basis gestellt wurden.

Entscheidend für den Nutzen von SOA ist hier wiederum ein gemeinsames Geschäftsverständnis, da SOA-Services nur Bausteine liefern; ohne Bauplan sind auch die interessanten Bausteine relativ nutzlos. Dieses gemeinsame Geschäftsverständnis wird durch branchenspezifische Prozessmodelle unterstützt, sozusagen der Top-down-Ansatz des Business-Engineering. Eine weitere Unterstützung entsteht derzeit dadurch, dass einzelnes Geschäfts-/Prozess-Wissen formal ausgewertet und zugänglich gemacht wird durch den Einsatz von Ontologien (semantische Informationen und Netze), sozusagen der Bottom-up-Ansatz des Business-Engineering.

Nur über ein harmonisiertes Verständnis und formale Nutzbarkeit der Geschäftsprozess-Beschreibung, der Taxonomie öffentlicher Services und auch vergleichbarer und vergleichender Service-Level-Beschreibungen wird eine erfolgreiche Beschreibung und Nutzung von öffentlichen Services im globalen Geschäftskontext möglich. Diese wichtige Aufgabe muss von Branchen-Verbänden, der IT-Industrie und der angewandten Informatik-Forschung gemeinsam bewältigt werden.

Ein zweiter wichtiger Aspekt ist die Koexistenz der „alten“ und der „neuen“ SOA-Welt. Besonders auch im Collaborative-Business-Bereich werden bestehende, gut funktionierende Lösungen nicht unbedingt durch etwas Neues abgelöst, nur weil es neu ist. Daher werden bestehende und neue Paradigmen vermutlich über längere Zeit nebeneinander und miteinander existieren. Erfahrung mit beiden Welten und ein Blick auf den Nutzen von SOA im Einzelfall sind hierbei wichtig.

Eine große Bedeutung haben natürlich Sicherheitsaspekte im öffentlichen Einsatz von SOA. Gerade die größere Flexibilität und Offenheit sind Fluch und Segen zugleich. Hier können aber viele Erfahrungen, die aus dem Internet-Umfeld gewonnen wurden, sehr gut zur Sicherung eingesetzt werden.

Die Sicherheit und verbindliche und akzeptierte Standards dazu sind für eine offene Technologie wie SOA sehr wichtig, weil ohne die Verlässlichkeit keine Akzeptanz gewonnen werden kann. Anerkannte technische Sicherheit sind

aber nur ein notwendiger Teil zur Akzeptanz von SOA. Vielfach herrschen geschäftliche oder auch emotional/persönliche Vorbehalte gegen einen offenen Umgang mit Informationen. Im SCM-Bereich etwa wird nach wie vor mit Informations-Politik Macht-Politik umgesetzt und technische Möglichkeiten stehen hier immer noch dem Wettbewerbskalkül gegenüber. Der Begriff „Collaborative Business" beschreibt daher oft mehr Vision als Wirklichkeit, weil man von einer Kollaboration (also Zusammenarbeit) entlang der Wertschöpfung oft noch weit entfernt ist. SOA stellt hierfür technische und organisatorisch wirksame Werkzeuge bereit; die Einsicht, dass ein globales Optimum entlang der Wertschöpfungskette besser sein kann, als viele lokale Optima wird vermutlich langsamer wachsen als die technische Umsetzungsfähigkeit.

Abschließend noch ein eigentlich trivialer Hinweis. Wertschöpfungsketten spannen sich heute global um die ganze Welt und dieser Aspekt der Internationalität ist natürlich auch für globale SOA wichtig. Nationale juristische, sprachliche aber auch finanzielle Aspekte können gravierende Hindernisse für globales Geschäft bilden. Globale SOA macht da keine Ausnahme, verstärkt möglicherweise den Druck in diese Richtungen und verschiebt somit den Leidensdruck von der IT-Umsetzung weg.

Fazit

SOA entlang der (globalen) Wertschöpfungsketten hat eine wichtige Bedeutung, stellt aber keineswegs eine Revolution im Collaborative Business dar, vielmehr einen Katalysator, der einen vorhandenen Trend beschleunigt. Die Analogie zeigt aber auch deutlich: Manche Reaktionen laufen ohne solche Katalysatoren nicht ab oder sind einfach ineffektiv und damit uninteressant.

Die Vorteile und der Nutzen von SOA im Collaborative Business sind:

- Eine gemeinsame Sprache aus der Perspektive des Geschäfts und nicht der technischen Implementierung (Geschäftsobjekte anstelle von Bits und Bytes)
- Der Einstieg ist technisch und vom Aufwand deutlich einfacher und flexibler als bei der klassischen Integration entlang der Wertschöpfungskette (XMA anstelle von EDIFACT)
- Der globale SOA-Ansatz wird durch interne SOA-Ansätze unterstützt (kein Paradigmen-Bruch zwischen „innen" und „außen")
- Durch die feinere Granularität und lose Kopplung ist SOA auch global flexibler und einfacher zu erweitern als die klassische Integration. Dies hilft überall dort, wo schnell und kreativ am Markt agiert werden soll oder muss.

3.6 Governance-Aufbau

SOA Governance – Einführung

Softwaresysteme haben sich in der Vergangenheit kontinuierlich in Bezug auf Größe, Verteilungsgrad und Komplexität weiterentwickelt, sodass heute hochkomplexe heterogene IT-Landschaften mit vielen unterschiedlichen Intergrationsansätzen das Bild in vielen Unternehmen prägen. Die IT und damit auch Unternehmensanwendungen sind in heutigen Unternehmen meist vertikal entlang der Geschäftsbereiche organisiert. Diese Tatsache verhält sich konträr zur Kernidee der SOA, die neben der Vereinheitlichung der Architektur auch das Ziel verfolgt, fachliche, geschäftsunterstützende Services bereichsübergreifend zur Verfügung zu stellen und mehrfach zu verwenden.

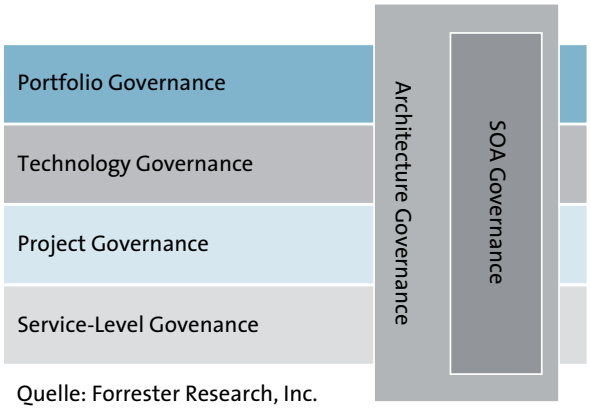
Während IT-Governance durchaus ein weitläufig benutzter und verstandener Begriff ist, trifft das für den Begriff „SOA Governance" noch nicht zu. Im Zusammenhang mit SOA führt der Begriff Governance immer wieder zu kontroversen Diskussionen, obwohl es sich dabei um eine der grundlegenden Voraussetzungen für eine langfristig erfolgreiche SOA handelt.

Der Begriff SOA Governance bezeichnet die Definition, Durchsetzung und Steuerung von organisatorischen Regeln, Richtlinien und Standards zur konsequenten Ausrichtung der Services und Geschäftsprozesse an der Unternehmensstrategie durch geeignete Steuerungs- und

Kontrollmaßnahmen. IT-Governance im konventionellen Sinne befasst sich hingegen mit der Organisation, Steuerung und Kontrolle der IT und der IT-Prozesse. Diese Aufgabe wird jedoch immer noch zu technisch gesehen. Durch die Einführung einer SOA rückt der Fokus auf Services und Geschäftsprozesse auch in den Mittelpunkt der IT-Governance.

Wie in Abbildung 11 dargestellt verhält sich SOA Governance orthogonal zu den klassischen Disziplinen der IT-Governance und ist nicht unmittelbar an einzelnen Projekten oder Anwendungen festzumachen. Sie ist vielmehr als bereichsübergreifende Disziplin und als integraler Bestandteil einer übergeordneten IT-Governance zu sehen.

Der Mehrwert von SOA Governance besteht darin, SOA-relevante Aspekte in den Vordergrund aller strategischen Initiativen zu rücken und eine Möglichkeit zu schaffen, diese unternehmensweit und einheitlich zu steuern. SOA Governance betrachtet das große Ganze – wie in Abbildung 12 dargestellt – und umfasst neben den technischen/technologischen Bereichen auch die geschäftsrelevanten Zweige. Der Fokus liegt auf der Verwaltung von Services und deren Eigenschaften im Laufe des Lebenszyklus mit dem Ziel, die Wiederverwendbarkeit von Services zu gewährleisten.



Quelle: Forrester Research, Inc.

Abbildung 11: IT-Governance

Portfolio Governance	▪ Application Portfolio Planning ▪ Business Service Portfolio Planning ▪ Project Portfolio Management
Technology Governance	▪ Strategic SOA Platform Planning ▪ Service Profile and Pattern Management
Project Governance	▪ Service Interface Design ▪ Service Implementation ▪ Service Policy Specification
Service-Level Governance	▪ Runtime Policy Enforcement ▪ Service-Level Management

Quelle: Forrester Research, Inc.

Abbildung 12: Fokusbereiche einer SOA Governance

Service-Eigenschaften

Ein Service lässt sich – wie jede andere Softwarekomponente auch – über zwei elementare Bestandteile definieren: funktionale und nicht funktionale Eigenschaften. Die funktionalen Eigenschaften beschreiben die fachliche Logik, die nicht funktionalen Eigenschaften dienen der Beschreibung weiterer teilweise ebenfalls geschäftsrelevanter Aspekte, wie z. B. Sicherheit, Verfügbarkeit, Skalierbarkeit in Bezug auf Performance. Die nicht funktionalen Eigenschaften spiegeln sich typischerweise in Service Level Vereinbahrungen (SLA) wider, während die funktionalen Eigenschaften in der Service-Implementierung umgesetzt werden. In der Vergangenheit wurden eine Vielzahl nicht funktionaler Eigenschaften im Service selbst implementiert, was ein Service-Wildwuchs zur Folge hatte. Je nach SLA mussten verschiedene Implementierungen derselben Schnittstelle bereitgestellt und verwaltet werden.

Erst eine klare Trennung zwischen funktionalen und nicht funktionalen Eigenschaften ebnet den Weg für eine Rollentrennung und somit auch für die Trennung der Verantwortlichkeiten. Sowohl die IT-Abteilung als auch der Fachbereich profitieren gleichermaßen von einer solchen Trennung: durch eine einzige Umsetzung (Programmierung) der fachlichen Logik und durch die Veröffentlichung einer einzigen Service-Schnittstelle lassen sich einerseits der Service-Wildwuchs in der IT verhindern (es existiert

nur eine einzige Service-Implementierung mit einer einzigen veröffentlichten Schnittstelle, die gewartet und gepflegt werden muss), und andererseits lässt die sinnvolle Wiederverwendbarkeit des Service erreichen (die Service-Implementierung lässt sich mit unterschiedlichen SLAs als Service-Angebot kombinieren).

Die nichtfunktionalen Eigenschaften sollten sich durch eine geeignete SOA Governance Lösung umgesetzen lassen.

Service-Lebenszyklus

Wie in Abbildung 13 illustriert folgen Services und Geschäftsprozesse einem Lebenszyklus, an dem verschiedene Bereiche und somit verschiedene Verantwortliche beteiligt sind.

Der Lebenszyklus eines Service oder eines Prozesses umfasst vier Kernbereiche:

- die Business-Domäne, die den Service/Prozess bereit stellt

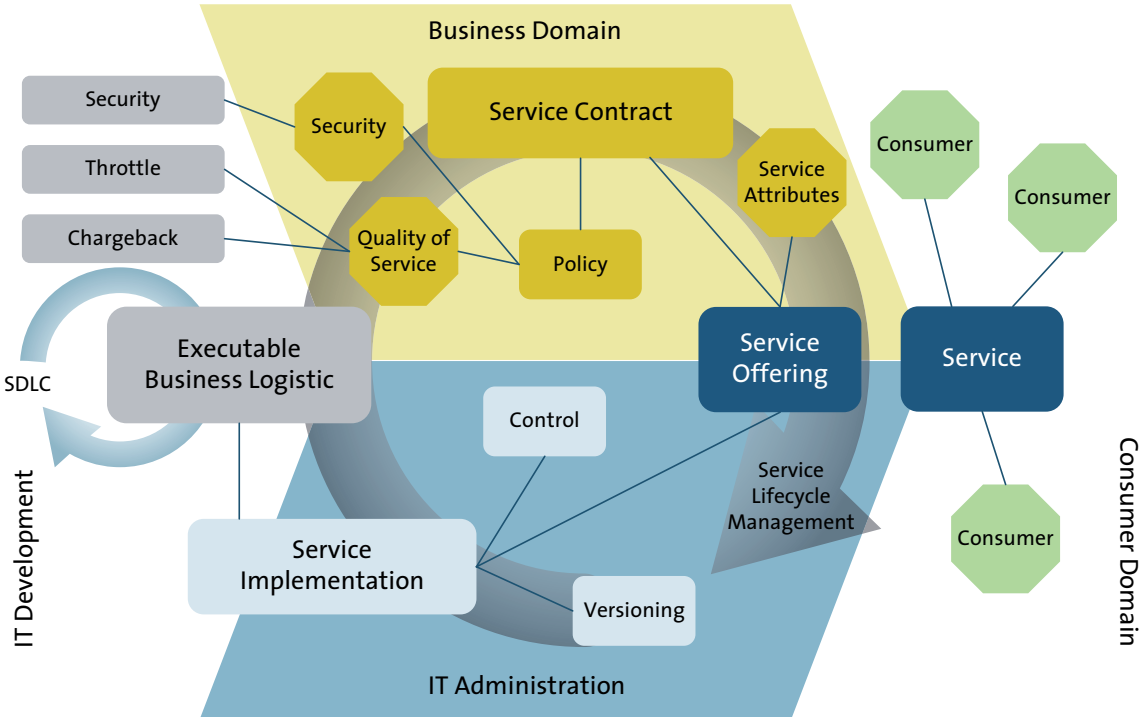


Abbildung 13: Service-Lebenszyklus und beteiligte Bereiche

- die (klassische) IT-Entwicklung (Architektur und Design, Entwicklung, Test)
- die (klassische) IT-Administration (Infrastruktur, Deployment, Betrieb)
- die Business-Domäne, die den Service/Prozess nutzt (Consumer-Domäne)

Im Rahmen eines SOA-Governance-Prozesses trägt jeder der o. g. Bereiche die Verantwortung für unterschiedliche Eigenschaften eines Service: Die IT-Entwicklung implementiert die fachliche Logik, die Business-Domäne definiert die funktionalen und nicht funktionalen Eigenschaften eines Service. Der IT-Betrieb ist schließlich für die Bereitstellung der Infrastruktur, den Betrieb und die Überwachung der Services verantwortlich.

Im Mittelpunkt einer Governance-Lösung stehen die Regeln und die Service-Level-Vereinbarungen, die im

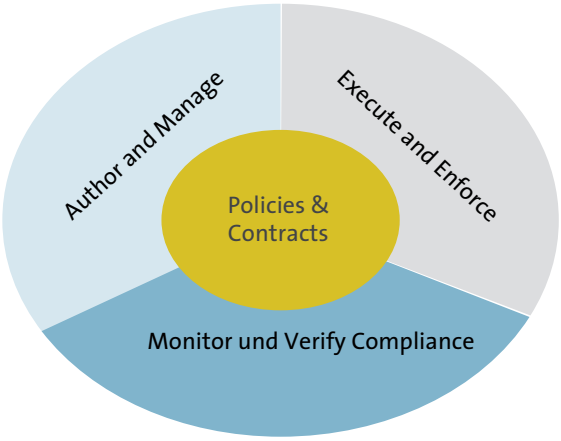


Abbildung 14: Aspekte einer SOA-Governance-Lösung

Rahmen des Governance-Prozesses aus drei unterschiedlichen Blickwinkeln betrachtet werden (siehe Abbildung 14). SOA Governance lässt sich in zwei Bereiche aufteilen: design-time Governance und run-time Governance.

Erstellung und Verwaltung

Dieser Bereich adressiert die Definition und Erstellung von Regeln, Richtlinien, Standards und Service-Level-Vereinbarungen. Services und die dazugehörigen Artefakte werden typischerweise in einer zentralen Service-Registry-/ einem zentralen Service-Repository verwaltet.

Ausführung und Durchsetzung

Während design-time Governance die Verwaltung von Service-Artefakten (Schnittstellendefinition, Deployment, SLA-Eigenschaften) im Blickpunkt hat, ist die run-time Governance für die Durchsetzung der festgelegten SLA-Aspekte zur Laufzeit der Services verantwortlich.

Steuerung durch Überwachung und Überprüfung

Die Einhaltung der festgelegten Regeln und Richtlinien lässt sich anschließend über eine zentrale Komponente überwachen und überprüfen.

SOA-Governance-Modell

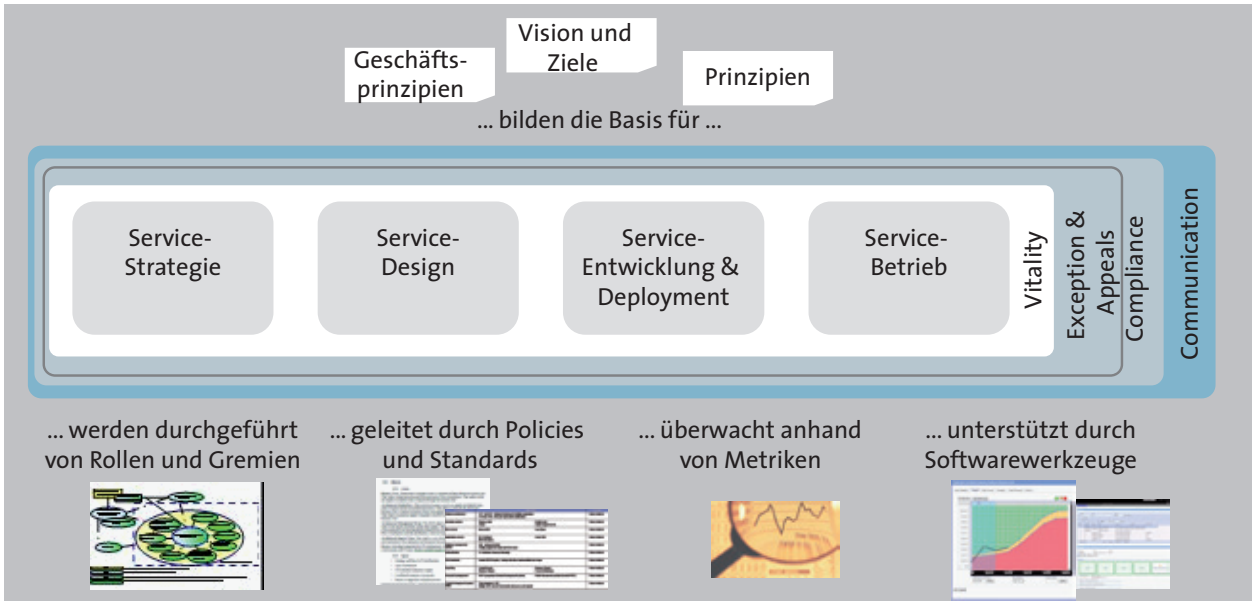


Abbildung 15: SOA-Governance-Modell

Wenn es aber bei SOA Governance um Entscheidungsrechte und Mechanismen zur Erfolgsmessung eines Service in seinem Lebenszyklus geht, stellt sich die Frage, wie und womit das geschieht. Ein Blick auf ein SOA-Governance-Modell (Abbildung 15) macht die Zusammenhänge deutlich.

Das Modell beinhaltet alle Elemente, die für die Steuerung von Services – durch ihren gesamten Lebenszyklus – erforderlich sind. Die Vision und Ziele im Kontext einer SOA sowie die daraus abgeleiteten Geschäfts- und IT-Prinzipien bilden die Basis für alle weiteren Aktivitäten im Lebenszyklus eines Service. Diese Aktivitäten sind repräsentiert durch die vier Prozessgruppen: Service-Strategie, Service-Design, Service-Entwicklung und -Deployment sowie Service-Betrieb.

Service-Strategie und Service-Design wurden bereits in diesem Kapitel beschrieben.

Lebenszyklus von Services

Das Konzept der Wiederverwendbarkeit von Services verursacht Abhängigkeiten zwischen den Fachabteilungen eines Unternehmens, da sie Anwendungsfragmente nutzen, die von jeweils anderen Abteilungen bereitgestellt werden. Bei der Änderung eines Service muss geprüft werden, ob die bisherigen Prozesse weiterhin operabel sind. SOA Governance beschäftigt sich deswegen auch mit dem Lebenszyklus von Services, damit die zwischen den Abteilungen vereinbarte SLAs eingehalten werden können. Die Lebensdauer eines Service ist endlich – die SOA Governance macht den Erstellungs-, Änderungs- und Auflösungsprozess transparent. Hierzu werden folgende Stufen definiert: Planung (Definition der Funktionalität und Design von Services), Überprüfung und Überarbeitung (Verhalten von neuen Services im operativen Alltag und deren Update), Deployment (Verwendung von aktiven Services) und Beendigung von Services. Insbesondere während der Stufe „Überarbeitung“ muss darauf geachtet werden, dass alle Unternehmensbereiche mit den veränderten Services ihr Tagesgeschäft erfüllen können.

Effizienzmessung und Betrieb

Im Rahmen einer Service-orientierten Architektur müssen Service-Level-Agreements bei der Bereitstellung eines Service erfüllt werden. Diese Vereinbahrungen gilt es zu überprüfen und zu messen, um eine gleichbleibend hohe Qualität der zur Verfügung gestellten Leistung zu garantieren. Hierzu dient die Effizienzmessung. Innerhalb eines Service werden sogenannte „key performance indicators“ bestimmt, also Anhaltspunkte, nach denen man einen Bewertung vornehmen kann. Aber nicht nur die Bewertung, sondern auch das Monitoring von Services spielt eine zentrale Rolle, sodass Probleme schon ausgeräumt werden können, ohne dass es zu einem Bruch der SLAs kommt. Mithilfe dieser Messung lässt sich darüber hinaus eine Kostenverteilung auf Basis des Verursacherprinzips für die IT-Infrastruktur auf die entsprechenden Kostenstellen durchführen.

Eine Messung der Key-Performance-Indikatoren verdeutlicht, welchen Einfluss ein verbessertes IT- Management, hinsichtlich der SOA-Thematik, auf das operative Geschäft eines Unternehmens hat. Gartner Research fand heraus, dass Projekte mit gehobenen Monitoring-Analysen schneller und bereitwilliger Geldgeber finden als Projekte ohne konsequentes Monitoring. Das Monitoring legt offen, welche Auswirkung der Einsatz von Services für das Tagesgeschäft eines Unternehmens hat.

Verschiedene Anbieter stellen Analyse-Tools zur Verfügung, die IT-Prozessketten, System-Performance und Services (bereitgestellt durch Applikationen) messen und dokumentieren können. Mithilfe dieser Anwendungen lassen sich Fehlerquellen ausfindig machen und Prozessverläufe simulieren.

Governance-Prozesse

Die Governance-Prozesse stellen sicher, dass die Aktivitäten im Lebenszyklus eines Service in Übereinstimmung

mit den definierten SOA-Zielen und Vorgaben durchgeführt werden. So werden die Ziele und Vorgaben durch entsprechende Kommunikation zu den Beteiligten bekannt gemacht. Die Einhaltung der Vorgaben wird kontinuierlich überwacht. Wenn Vorgaben aus bestimmten Gründen nicht eingehalten werden können, so werden Ausnahmegenehmigungen erteilt und verwaltet. Regelmäßige Überprüfungen und Analysen passen zudem das Governance-Modell an geänderte Rahmenbedingungen an.

Der Etablierung von Gremien zur Durchführung dieser notwendigen Aktivitäten und Prozesse, die durch den Governance-Prozess sowie durch den Lebenszyklus eines Service vorgegeben sind, ermöglicht die reibungslose Umsetzung im Unternehmen. Richtlinien und Standards geben vor, wie diese Prozesse abzuarbeiten sind. Eine entsprechende Erfolgsmessung erfolgt kontinuierlich mithilfe von geeigneten Metriken. In einigen Kontexten werden die Aktivitäten durch entsprechende Softwarewerkzeuge unterstützt, z. B. durch eine Service-Registry bzw. ein -Repository.

Hat ein Unternehmen bereits erfolgreich andere Governance-Mechanismen etabliert, kann die SOA Governance auf die bestehenden Mechanismen und Strukturen aufbauen. Beispielsweise werden neue Rollen oder Standards für Servicebeschreibungen ergänzt.

Fazit

- SOA Governance bezeichnet die Definition, Durchsetzung und Steuerung von organisatorischen Regeln, Richtlinien und Standards zur konsequenten Ausrichtung der Services und Geschäftsprozesse an der Unternehmensstrategie durch geeignete Steuerungs- und Kontrollmaßnahmen.
- SOA Governance verhält sich othogonal zur klassischen IT-Governance und ergänzt diese um Service-orientierte Aspekte
- SOA Governance berücksichtigt und unterstützt den Lebenszyklus von Services und Geschäftsprozessen
- Ohne ein geeignetes SOA-Governance-Modell ist die Einführung zum Scheitern veruteilt

3.7 Strategie und Taktik

Strategische und taktische Planung und Umsetzung von SOA-Projekten

Geht es um die Implementierung von SOA-Vorhaben, stehen viele Unternehmen und deren IT-Abteilungen vor einer Herausforderung ähnlich der Quadratur des Kreises: Ohne Beeinträchtigung des laufenden Betriebes sollen eine neue Technologie und neue Systeme eingeführt, alte Systeme stillgelegt und Datenbestände migriert werden – selbstverständlich bei gleichzeitig gesteigerter Agilität und zu geringeren Kosten. Bei unzureichender Planung lässt sich dieser Zielkonflikt kaum entschärfen! Verschiedene strategische und taktische Transformationsansätze – je nach Ausgangslage im Unternehmen und angestrebten Zielen – versprechen jedoch Erfolg.

Eines der größten Probleme bei der Umsetzung von SOA-Vorhaben besteht in der Tatsache, dass im Allgemeinen nicht bei Null gestartet wird, sondern vielmehr bereits eine Systemlandschaft existiert, die häufig historisch gewachsen, technisch und architektonisch veraltet und mit vielen Problemen beladen ist. Nicht selten sind sogar die Probleme mit der alten Systemlandschaft die treibende Kraft hinter der SOA-Initiative – in der Hoffnung auf eine „schöne neue Welt“.

Für eine verständlichere Darstellung der Planungsansätze sei eine fiktive Ausgangssituation mit folgenden Merkmalen beschrieben:

- Heterogene Systemlandschaft
- Viele Punkt-zu-Punkt-Integrationsschnittstellen
- Einsatz unterschiedlicher, größtenteils proprietärer Integrationstechnologien und Protokolle
- Starke technische und funktionale Abhängigkeiten der Systeme untereinander
- Keine Integrationsplattform im Einsatz

Je nach Ausgangslage und individuellen Prioritäten können taktische oder strategische Ziele die Oberhand gewinnen. Dementsprechend muss auch der SOA-Transformationsprozess an diese übergreifenden Ziele angepasst werden.

Taktisch (Fokus auf kurzfristigen Vorteilen)	Strategisch (Fokus auf langfristigen Zielen)
möglichst kurzfristiger ROI	langfristige Unternehmensziele
Verbesserung/Upgrade bestehender Systeme	Anschaffung neuer Systeme
Notwendigkeit, kurzfristig neue Systeme einzuführen	Planung und Umsetzung einer unternehmensweiten Applikationslandschaft
Optimierung bestehender Prozesse	Implementierung neuer Prozesse und Geschäftsfelder
Optimierung bestehender Prozesse	Implementierung neuer Prozesse und Geschäftsfelder

Tabelle 1: Gegenüberstellung taktischer und strategischer Transformationsansatz

Der taktische Transformationsansatz im Detail

Der taktische Ansatz zielt zunächst auf eine kurzfristige Verbesserung der Ausgangslage, indem zuerst die bestehende Applikationslandschaft durch Einführung einer Enterprise-Application-Integration-(EAI-)Plattform entflochten wird. Die daraus resultierende Verringerung der Anzahl der Schnittstellen in Verbindung mit den verbesserten Überwachungs- und Managementfunktionen einer EAI-Plattform führt in der Regel schon kurzfristig zu einer Senkung der laufenden Betriebskosten. Gleichzeitig wird der kommenden SOA der Weg bereitet, da im nächsten Schritt der Zielzustand durch eine Aufwertung der EAI-Plattform hin zu einem Enterprise Service Bus (ESB) auf einem relativ geradlinigen Weg erreicht werden kann. Allerdings ist für ein solch schrittweises Vorgehen auch mehr Zeit einzuplanen und Kompromissbereitschaft nötig, da die bestehende Applikationslandschaft im Kern beibehalten wird. Eine radikale Umgestaltung ist auf diesem Wege nicht zu erreichen.

Das schrittweise Vorgehen:

- Unterteilung der bestehenden Systemlandschaft in unabhängige Subsysteme. Ein Subsystem besteht dabei aus einer Gruppe von Einzelsystemen, die gemeinsam eine klar abgegrenzte Funktion haben, untereinander eng gekoppelt sind, aber in ihrer Gesamtheit nur eine oder wenige Schnittstellen nach außen hin aufweisen.
- Implementierung und Test einer EAI-Plattform mit den erforderlichen Adaptern zur Anbindung der einzelnen Subsysteme. Anmerkung: Bis zu diesem Zeitpunkt bleibt die bestehende produktive Systemlandschaft unangetastet. Die EAI-Plattform wurde mittels Testsystemen getestet und steht nun in der Produktivumgebung bereit.
- Die Subsysteme werden nun schrittweise mit der EAI-Plattform verbunden; die alten Punkt-zu-Punkt-Verbindungen werden dabei aufgelöst. Wichtig: bei diesem Schritt ist es nicht notwendig, umfangreiche Änderungen an den bestehenden Systemen vorzunehmen. Die bestehenden Schnittstellen werden sozusagen nur „neu verdrahtet“, ohne dass die angeschlossenen Systeme davon etwas merken. Unterschiedliche Protokolle und Datenformate werden von der EAI-Plattform umgesetzt. Im Einzelfall lassen sich kleinere Änderungen jedoch nicht vermeiden. So kann es beispielsweise sein, dass ein System zusätzliche Informationen an seiner Schnittstelle bereitstellen muss, damit die EAI-Plattform die Informationen an das richtige Zielsystem weiterleiten kann.
- Im nächsten Schritt lassen sich die durch die Entflechtung gewonnenen Freiheitsgrade zur Modernisierung

der bestehenden Systemlandschaft nutzen. Die Aktualisierung oder Ablösung von Altsystemen und die Einführung neuer Systeme fällt nun deutlich leichter, da immer nur eine Schnittstelle (zur EAI-Plattform) betroffen ist.

- Parallel dazu kann mit dem Ausbau der EAI-Plattform zu einem ESB das SOA-Zeitalter eingeläutet werden. Konkret bedeutet das: Einführung von Web Service Standards (WS-*); sukzessive Ablösung bestehender proprietärer Schnittstellen durch Web Services. In dieser Übergangsphase muss die EAI-Plattform die Brücke zwischen der neuen und der alten Welt aufrechterhalten, indem sie die proprietären Protokolle der Altsysteme in Web-Service-Protokolle (und umgekehrt) umsetzt.
- Nun kann auch zügig mit der Implementierung von unternehmensweiten Diensten begonnen werden, denn erst durch die gemeinsame Nutzung und Wiederverwendung von Diensten entfaltet eine SOA ihre größten Vorteile. Wenn am Ende alle Systeme ausschließlich Web Services nutzen und auch bereitstellen, ist die SOA-Transformation abgeschlossen.

Der strategische Transformationsansatz im Detail

Der strategische Ansatz verzichtet zunächst auf kurzfristige Erfolge und setzt daher eine gewisse Investitionsbereitschaft voraus. Er beginnt mit der Definition der Zielarchitektur, ohne die bestehende Applikationslandschaft einzubeziehen, und führt somit zu einer wesentlich radikaleren Umgestaltung, die hier allerdings auch gewünscht ist. Bei gründlicher Planung und Vorbereitung können bei dieser Vorgehensweise viele Aufgaben parallel in Angriff genommen werden und demnach insgesamt schneller zum Ziel führen als ein von taktischen Notwendigkeiten geprägter Ansatz.

Das schrittweise Vorgehen:

- Entwurf der Zielarchitektur und Applikationslandschaft auf Basis der Anforderungen des (zukünftigen) Geschäftsmodells. Die Applikationslandschaft wird dabei aufgeteilt in unternehmensweite Basisdienste (wie zum Beispiel Dokumentenmanagement, Customer Data Integration, Referenzdatenmanagement, etc.), die von allen Abteilungen und Geschäftsbereichen gleichermaßen genutzt werden können, und geschäftsspezifischen Diensten (beispielsweise Rechnungswesen, Auftragsbearbeitung, etc.).
- Nun wird versucht, die bestehende Applikationslandschaft auf die neue Ziellandschaft abzubilden, indem nach existierenden, die gewünschten Dienste der Ziellandschaft bereits bereitstellenden Applikationen gesucht wird. Dabei werden jedoch diejenigen Systeme außer Acht gelassen, die aus technischen oder funktionalen Gründen bereits zur Ablösung oder Stilllegung vorgesehen sind. Das Ergebnis weist in der Regel einige nicht unerhebliche Lücken auf, die mit neuen Applikationen besetzt werden müssen.
- Planung, Implementierung und Test einer ESB-Plattform inklusive aller technischen Basisdienste (Service Directory, Metadata Repository, Rules Engine, etc.) sowie der dazugehörigen Überwachungs- und Analysemöglichkeiten.
- Bereitstellung der unternehmensweiten Basisdienste aus der Ziel-Applikationslandschaft als Web Services im ESB. Anmerkung: Bis zu diesem Zeitpunkt bleibt die bestehende produktive Systemlandschaft unangetastet. Die ESB-Plattform wurde mit Testsystemen getestet und steht nun in der Produktivumgebung bereit.
- Jetzt kann mit der eigentlichen SOA-Transformation begonnen werden. Die bestehenden Systeme werden – nach und nach oder gleichzeitig – mit Web-Service-Schnittstellen versehen und an den ESB angebunden und können nun damit beginnen, von den Vorteilen der bereits vorhandenen Basisdienste zu profitieren.
- Parallel dazu können die als ungeeignet eingestuften alten Applikationen durch neue, service-orientierte Applikationen abgelöst werden. Wurde die Ziel-Applikationslandschaft am Ende vollständig auf der Basis von service-orientierten Applikationen umgesetzt, lässt sich auch hier die SOA-Transformation als abgeschlossen betrachten.

Taktischer SOA-Transformationsansatz	Strategischer SOA-Transformationsansatz
Vorgehensweise: ■ Unterteilung in Subsysteme ■ EAI-Plattform ■ Anschluss der Subsysteme ■ Basisdienste ■ ESB-Upgrade ■ SOA-Upgrade/Migration der Einzelsysteme	Vorgehensweise: ■ Definition der Ziel-Applikationslandschaft ■ ESB-Plattform ■ Basisdienste ■ SOA-Anschluss/Migration der Einzelsysteme
Geeignet für:Unternehmen mit einem relativ stabilen Geschäftsmodell und einer im Großen und Ganzen funktionierenden Applikationslandschaft, aber einer komplexen, historisch gewachsenen Integrationslandschaft.	Geeignet für: Unternehmen, die sich gerade neu ausrichten oder deren Applikationslandschaft den Anforderungen nur ungenügend gerecht wird, was eine Neugestaltung notwendig macht.

Tabelle 2: Zusammenfassung der SOA-Transformationsansätze

4 Architektur und Aufbau

4.1 Mapping: Business und IT

Geschäft und IT – Notwendigkeit der Verzahnung

Ein wesentliches Merkmal von SOA ist die enge Abstimmung und Zusammenarbeit von Geschäft und IT. Das SOA-Konzept und dessen Methoden sind die Grundlage für diese enge Verzahnung.

Unternehmen sichern ihre Wettbewerbsfähigkeit durch die schnelle Reaktion auf sich ändernde Marktsituationen und Kundenanforderungen. Hierfür bedarf es eines geeigneten Geschäftsmodells. Ein solches Geschäftsmodell leitet sich aus der Analyse der unternehmerischen Möglichkeiten sowie dem Erkennen neuer Geschäftschancen ab.

Organisation und Prozesse eines Unternehmens müssen so ausgerichtet sein, dass sie das Geschäftsmodell und die daraus folgenden Anforderungen optimal unterstützen. Die Prozesse müssen so gestaltet werden, dass sie auf Änderungen des Geschäftsmodells schnell angepasst werden können.

Die optimale Unterstützung durch die Organisation und die Prozesse setzt voraus, dass die gesamte IT und einzelne Anwendungen die an sie gerichteten Anforderungen erfüllen. Geschäft und IT müssen also eng ineinandergreifen: Geschäftsanforderungen müssen definiert werden, IT-Anforderungen müssen entsprechend abgeleitet werden und schließlich müssen diese Anforderungen mit den vorhandenen IT-Potenzialen abgeglichen werden.

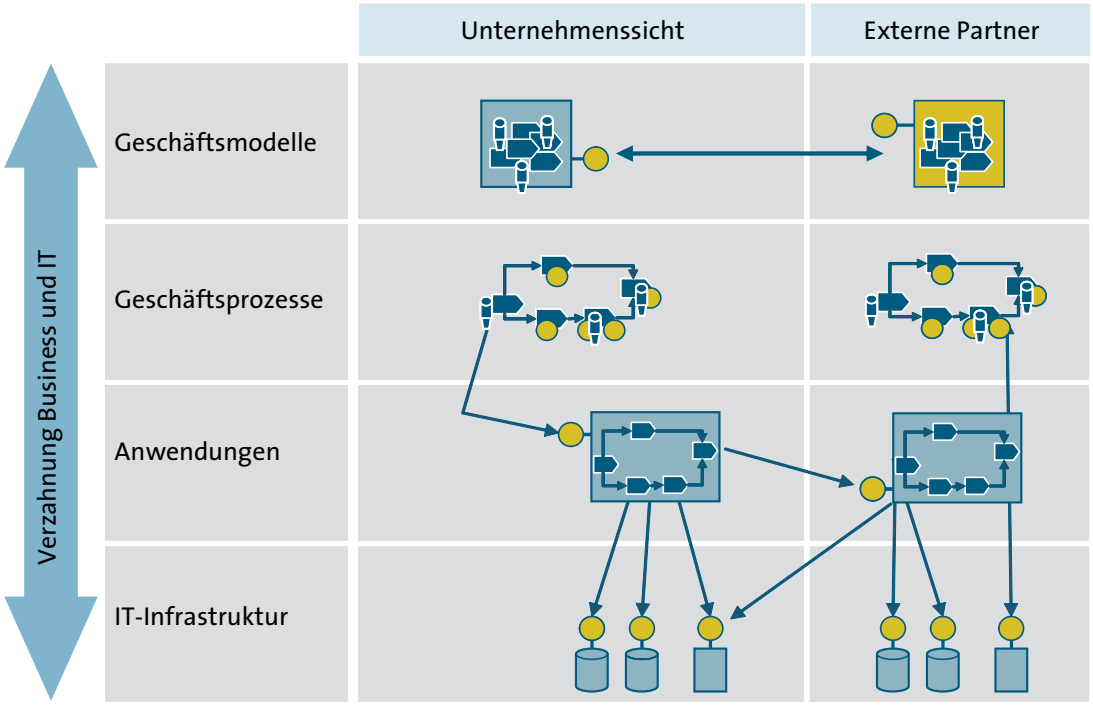


Abbildung 16: Verzahnung Business und IT

Verzahnung Geschäft und IT – das Vorgehen

Abbildung 16 zeigt die Interaktion und Abhängigkeiten der verschiedenen Bereiche, die bei der Beschreibung von Anforderungen, deren Umsetzung sowie bei Änderungen zu berücksichtigen sind, die im laufenden Betrieb auftreten. Ein systematisches Vorgehen bei der Erstellung und ggf. Änderung von Services stellt sicher, dass diese Abhängigkeiten in jeder Phase des Projektes berücksichtigt werden.

Die Verzahnung nach dem Top-down-Ansatz setzt häufig auf der Ebene des Geschäftsmodells an: Dort wird die Ausrichtung des Unternehmens festgelegt – abhängig von der Unternehmensstrategie sowie der Markt- und Wettbewerbssituation (z. B. auch unter Einbeziehung von externen Partnern).

Auf der Geschäftsprozessebene werden die Abläufe definiert, die erforderlich sind, um das neue bzw. geänderte Geschäftsmodell umzusetzen. Die Zielsetzung der neuen Prozesse, also die Anforderungen, die an diese gestellt werden, ergeben sich aus dem Geschäftsmodell und dessen einzelnen „Komponenten". Bei der Validierung der Ergebnisse ist es wichtig, die ursprüngliche Zielsetzung heranzuziehen.

Eine wichtige Anforderung an die Gestaltung von Prozessen ist es, dass diese schnell an veränderte Geschäftssituationen angepasst werden können: Beispiele hierfür sind geänderte Kundensegmente, Wechsel und Anbindung von verschiedenen Partnern. Bei der Prozessdefinition muss festgelegt werden, welche Elemente fest in den Prozessablauf verankert werden und welche „lose" gestaltet werden. Stärkere Flexibilitätsanforderungen bzw. erwartete Änderungshäufigkeit führen zu einer Verkettung von „losen Elementen", den sogenannten Business Services. Nach der Definition der erforderlichen Prozesse und Services werden die erforderlichen IT-Services identifiziert.

In diesem Kontext wird nach dem Bottom-up-Prinzip vorgegangen, d. h., es wird eine Bestandsaufnahme der verfügbaren und erzeugbaren Services aus dem bestehenden IT-Umfeld heraus vorgenommen.

Da das Identifizieren und Definieren der Services die Schlüssel für eine erfolgreiche SOA-Einführung sind, empfiehlt es sich dabei auf bewährte Methoden und Verfahren zurückzugreifen.

Ein Abgleich der neu definierten Services und der bereits verfügbaren Services ergibt die neu zu erstellenden Services. Anschließend wird analysiert, wie die Services bereitgestellt werden können. Es ergeben sich folgende Möglichkeiten:

- Anpassung der bestehenden Anwendungen
- Einkauf der Services bei externen Service-Anbietern
- Neuentwicklung der Services

Im Rahmen der oben genannten Analyse werden auch die wirtschaftlichen Aspekte berücksichtigt. Dies kann zu einer Neu-Definition der Prozesse führen. Durch dieses Vorgehen wird sichergestellt, dass die von der IT bereitgestellten Funktionalitäten den geschäftlichen Anforderungen entsprechen.

Hier die wesentlichen Aktivitäten des beschriebenen Vorgehens:

- (1) Geschäftsmodell definieren
- (2) Anforderungen an die Geschäftsprozesse, IT und Organisation ableiten
- (3) Geschäftsprozesse definieren
- (4) Business Services identifizieren
- (5) Bestehende Anwendungs- und IT-Landschaft analysieren
- (6) Verfügbare Services identifizieren
- (7) Angeforderte Services mit bestehenden Services abgleichen
- (8) Spezifizierung und Realisierung der Services planen

Die Abschnitte 3.6 und 5.2 (Governance) beschreiben die Methoden und deren Umsetzung, die für eine umfassende Abstimmung von Business und IT unerlässlich sind.

Besonders hervorzuheben ist dabei das SOA Center of Excelence, welches den Rahmen für die Zusammenarbeit definiert, sicherstellt und unterstützt.

Organisatorische Empfehlung

Für eine erfolgreiche Integration von Business und IT ist es erforderlich, dass das Einführungsprojekt (Definition der Anforderungen) sowie der gesamte Lebenszyklus von Services (inkl. Betrieb) von Mitarbeitern aus den Fachbereichen und der IT gemeinsam durchgeführt bzw. begleitet wird.

Es empfiehlt sich zu Beginn der „SOA-Reise“ eine gemeinsame Sprache festzulegen, die sowohl der Fachbereich als auch die IT beherrschen und somit kommunikative Hürden vermieden werden.

Kritische Erfolgsfaktoren

- Organisation
  Jedem Prozess und Service wird ein Besitzer (Owner) zugeordnet, der bei der Definition aber auch im Betrieb Ansprechpartner ist und Entscheidungsbefugnis hat.
- Organizational Change Management
  Um die Akzeptanz und Unterstützung der Organisation bei der Einführung und Umsetzung des beschriebenen Vorgehens zu sichern, empfiehlt es sich ein begleitendes Organizational Change Management aufzusetzen.
- Governance Framework
  Legt die Rollen der Prozess- und Service-Owner fest, definiert das Vorgehen und die Methoden zur Sicherstellung der Einhaltung derselben (siehe auch Governance-Abschnitte in diesem Dokument)
- Korrekte Modellierung
  Wie die Modellierung zu erfolgen hat, damit alle Aspekte (organisatorisch, KPI, IT) abgedeckt sind und so beschrieben werden, dass die nachfolgenden Aktivitäten nahtlos daran anschließen können, wird durch das CoE o. ä. festgelegt.
- Tools
  Um Aufwand zu vermeiden, sollten die eingesetzten Werkzeuge sicherstellen, dass die Ergebnisse der jeweils vorangegangenen Aktivität übernommen und weiterbearbeitet werden können, insbesondere die Integration von Business und IT muss von ihnen unterstützt werden.
- Enterprise Architecture
  (Bestehende) Enterprise-Architecture-Konzepte sollen auch im Rahmen von SOA weiterverfolgt werden, da hier bereits eine Basis für die Verlinkung zwischen Business und IT besteht. EA-Artefakte liefern einen Überblick der Sicht aller Beteiligten.

4.2 Architektur-Entwurf

Dieses Kapitel beschreibt den Entwurf einer Service-orientierten Architektur. Der erste Teil illustriert anhand einer Referenzarchitektur, welche Elemente eine vollständig ausgebaute Service-orientierte Architektur enthalten sollte. Im Anschluss daran werden anhand verschiedener Szenarien Möglichkeiten der Abbildung auf konkrete Umgebungen aufgezeigt, damit wird auch die Bedeutung der Architekturelemente für unterschiedliche Abstraktionsebenen erklärt. Der dritte Teil beschäftigt sich mit der Frage, wie viel Geschäftslogik ein Dienst verkörpert (Granularität) und nach welchen Kriterien Prozesse durch Dienste abgebildet werden sollten.

Dieser Leitfaden bietet eine technologieunabhängige Sicht auf SOA und setzt daher seinen Fokus auch auf entsprechende Standards. Den Standard für die technische Umsetzung einer SOA stellen die Web-Services-Aktivitäten des World Wide Web Consortium (W3C), mit denen sich das Kapitel 7 beschäftigt.

Der letzte Abschnitt dieses Kapitels grenzt SOA von Ereignis-orientierten Architekturen ab (EDA) bzw. nachrichtenbasierten Architekturen (MOM) ab und zeigt, dass sich die unterschiedlichen Ansätze nicht ausschließen müssen, sondern sich ergänzen können.

SOA-Referenzarchitektur

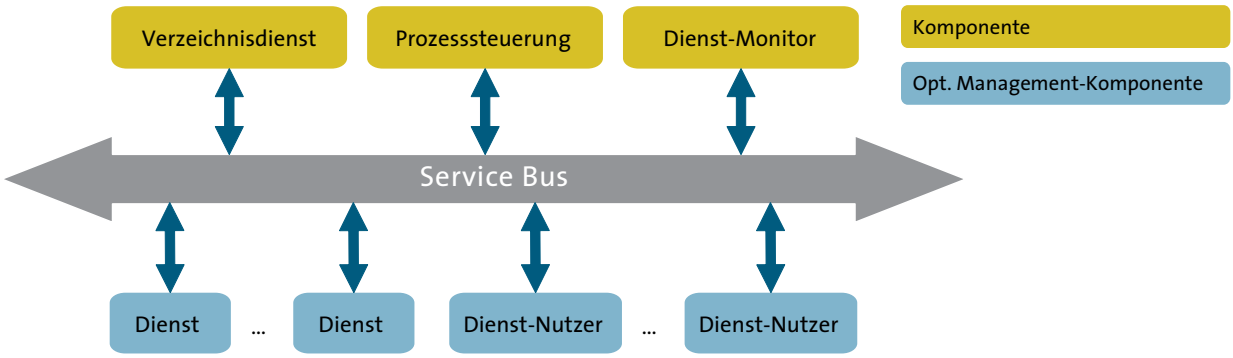


Abbildung 17: Komponenten in einer Service-orientierten Architektur

Komponenten

In unserer SOA-Referenzarchitektur unterscheiden wir nach ihren jeweiligen Aufgaben Komponenten, die in der folgenden Tabelle nach ihren spezifischen Eigenschaften charakterisiert werden. Für einige Komponenten werden in der Beschreibung alternative Ausprägungen angeboten, weil die konkreten Umsetzungen von den Anforderungen abhängen.

Der Dienst („Service“)

Der Dienst ist die funktionale Komponente einer SOA und stellt dabei die fachliche Funktionalität bereit. Er enthält eine oder mehrere Schnittstellen, die die fachlichen Prozesse in Form von Aktivitäten (Aufforderung, etwas „zu tun“) oder Ereignissen (es ist etwas „passiert“) als „Vertrag“ definieren.

Beispiel: ein Auftragsdienst stellt alle Aktivitäten (wie „weise Auftrag zu“) und Ereignisse (wie „Auftrag erzeugt“) aus dem Bestellprozess zu einem Auftrag bereit.

Dienstschnittstellen sollten generell versionsbezogen bereitgestellt werden, damit ein Dienst einen geregelten Lebenszyklus erhält und nicht alle Nutzer Anpassungen durchführen müssen, wenn eine Schnittstelle eines Dienstes verändert wird. Die Schnittstellen sind unabhängig von der Implementierung unseres Dienstes, sodass ein Dienst mehrere Versionen seiner Schnittstellen gleichzeitig bereitstellen kann.

Der Dienstnutzer („Service Consumer“)

Der Dienstnutzer nutzt bereitgestellte Dienste. Beispielsweise kann eine Portalanwendung einen Auftragsdienst verwenden, um den aktuellen Stand der Aufträge kundengerecht darzustellen.

Der Verzeichnisdienst („Repository“ und „Registry“)

Der Verzeichnisdienst stellt alle Informationen bereit, die zum Auffinden und zur Nutzung eines Dienstes benötigt werden, u. a.:

- Zieladresse und Protokoll des zu verwendenden Dienstes
- weitere Metainformationen (Metadatenmanagement) zur Beschreibung des Dienstes
- SLA-Informationen
- Zustand des Dienstes („Notbetrieb“, „o. k“, ...)

Hierbei unterscheidet man die Bereitstellung von Informationen zur Entwicklungs- und zur Laufzeit. Mit beidem zusammen ist es möglich, die Lebenszyklen bzw. den

Einsatz der Dienste zu steuern (Lifecyclemanagement) und zu überwachen (Governance).

Zur Entwicklungszeit werden den Entwicklern, Architekten, Analysten u. v. m. Informationen über die Funktionalität, das Design und ggf. die Rahmenbedingungen zur Verwendung der Dienste angeboten.

Zur Laufzeit ist es für den Betrieb der Dienste notwendig zu wissen, wo welche Dienste in welcher Version im Einsatz sind, wer sie benutzt und benutzen darf oder in welchem Zustand sie sind (z. B. Notbetrieb eines Dienstes).

Der Service-Bus

In komplexen oder größeren SOAs existieren sehr viele Dienste. Hier empfiehlt es sich, Dienste nicht direkt miteinander, sondern über einen Service-Bus miteinander kommunizieren zu lassen. Der Service-Bus unterstützt über Konnektoren vielfältige Transportprotokolle, sodass er z. B. zwei Dienste miteinander verbindet, bei denen der eine über http und der andere via MQ kommuniziert.

Darüber hinaus stellt ein Service-Bus üblicherweise weitere Leistungen zum Betrieb der Dienste bereit wie Logging, Auditing, Sicherheit, Nachrichten Transformation oder Transaktionsmanagement.

Die Prozess-Steuerung (Process-Engine)

Um einen Prozess abzubilden, müssen die einzelnen Aktivitäten und Ereignissen zu Abläufen, dem Prozess, zusammengesetzt werden. Das können auch Teilprozesse sein. Die Modellierung kann grafisch zur Entwicklungszeit über ein geeignetes Prozessmodellierungswerkzeug erfolgen. Die daraus resultierende Prozessdefinition dient dann der Steuerung des Prozesses, also der Auslösung der einzelnen Aktivitäten bzw. Ereignisse der Dienste.

Die Steuerung kann dabei

- direkt erfolgen (Orchestrierung): Die Prozess-Steuerungskomponente ruft den Dienst auf.
- indirekt erfolgen (Choreographie): Die Prozess-Steuerungskomponente verwaltet die Prozessdefinition und den Zustand des Prozesses und gibt z. B. einem Dienst vor, welche Aktivität bzw. welches Ereignis als nächstes auszulösen ist.

Der Dienst-Monitor

Dadurch dass die Aktivitäten bzw. Ereignisse außerhalb des Dienstes ausgelöst bzw. deren Beendigung sichtbar werden, ist das Verhalten messbar und damit auch überwachbar. Anhand der Prozessdefinitionen kann das Verhalten auch nicht funktional (z. B. Geschwindigkeit, Häufigkeit, ...) in Form von SLAs definiert werden. Das ist die Aufgabe eines Dienst-Monitors, die Parameter werden idealerweise durch den Verzeichnisdienst verwaltet und bereitgestellt. Damit werden die Aktivitäten und Ereignisse zu messbaren Leistungen!

Beispiel: Der Auftragsdienst stellt die Aktivität „Neuen Auftrag Anlegen“ bereit. Er soll zwischen 8:00 und 20:00 Uhr verfügbar sein. Es sollen bis zu 500 Aufträge/h angelegt werden. Eine Anlage darf nicht länger als eine Sekunde dauern. Diese Parameter sind im Verzeichnisdienst hinterlegt. Der Dienst-Monitor überwacht die Anfragen an den Dienst („Neuen Auftrag Anlegen“) und prüft anhand der Antworten die obigen Parameter. Damit kann er auch Fehler identifizieren (keine Antwort).

Die transparente Abbildung der Geschäftsprozesse

Eine erfolgreiche SOA sollte die einheitliche und bestmögliche Abbildung von Geschäftsprozessen sichern. Deswegen sollte es auch Bestandteil einer erfolgreichen SOA sein, Abbildungsvorschriften zu definieren – und zwar:

- wie die Dienste definiert werden (s. o.)
- wie die Geschäftsprozesselemente (Aktivitäten und Ereignisse) in die Dienste abgebildet werden

Dann ist es sogar möglich, diese Abbildungen zu automatisieren, also aus den Prozessdefinitionen, Dienstschnittstellen zu generieren. MDA und SOA können sich also sinnvoll ergänzen.

Unabhängig von der Vorgehensweise ist die bestmögliche Unterstützung des Geschäfts das Ziel einer SOA, das sich durch die Geschäftsprozesse und den daran beteiligten Menschen definiert. „Top-down“ und „Bottom-up“ sollten damit in beiden Fällen die Definition des Sollprozesses realisieren. Während der Top-down-Ansatz die Prozessdefinition als Ausgangspunkt bzw. Vorlage benutzt, sollte sich beim Bottom-up-Prozess der Soll-Prozess ergeben. Ansonsten müsste der Soll-Prozess, der dann zum Ist-Prozess wird, angepasst werden. Im Folgenden wird der „Top-Down“ Ansatz betrachtet. Hier gibt es viele Entwurfsmöglichkeiten, sodass der Leitfaden hier nur eine Empfehlung darstellen kann.

Ausgangspunkt: Die Prozessdefinition

Aus dem Ziel lässt sich die Anforderung ableiten, dass die Elemente der Geschäftsprozesse auf die technischen Operationen der Dienstschnittstellen so genau wie möglich abgebildet werden müssen.

Die gängigsten, fachlichen Beschreibungen von Geschäftsprozessen sind die ereignisorientierten Prozessketten (EPKs) und Sequenzdiagramme bzw. UML-Modelle.

Hier unterscheidet man Aktivitäten von Ereignissen.

Bei einer Aktivität, fordert ein Akteur (z. B. eine Organisationseinheit) einen anderen Akteur (z. B. eine andere Organisationseinheit) auf, etwas zu tun.

Beispiel: Der Vertrieb fordert das Auftragsmanagement auf, einen neuen Auftrag anzulegen.

Bei einem Ereignis kommuniziert ein Akteur (z. B. eine Organisationseinheit), dass sich etwas ereignet hat. Der Zustand eines Prozesses hat sich also geändert. Auf dieses Ereignis können andere Akteure mit Aktivitäten reagieren.

Beispiel: Das Auftragsmanagement signalisiert (das Ereignis), dass ein Auftrag erzeugt worden ist. Daraufhin reagiert der Vertrieb, indem er mit dem Kunden Kontakt aufnimmt und ihm mitteilt, dass der Auftrag angenommen worden ist.

Betrachtet man die Kommunikation zwischen zwei Akteuren, ist es also besonders wichtig, darauf zu achten, ob ein Akteur aufgefordert wird, etwas zu tun (Aktivität) oder ob er auf ein Ereignis reagiert und selbst entscheidet,

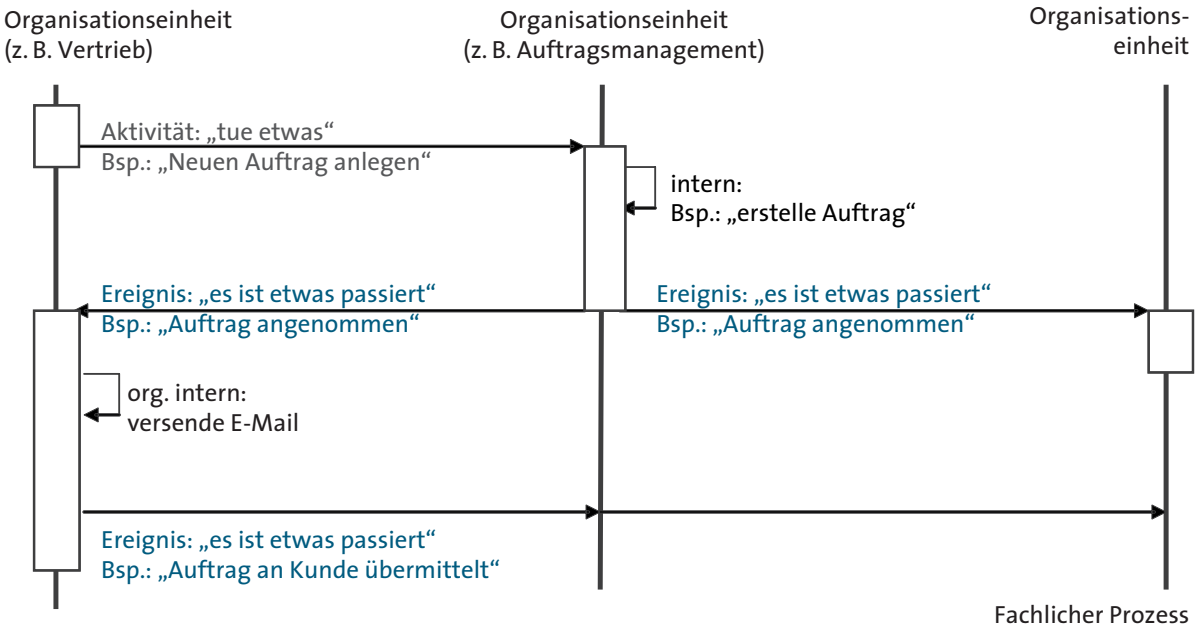


Abbildung 18: Abbildung einer Prozessdefinition mit Aktivitäten und Ereignissen in UML-naher Darstellung

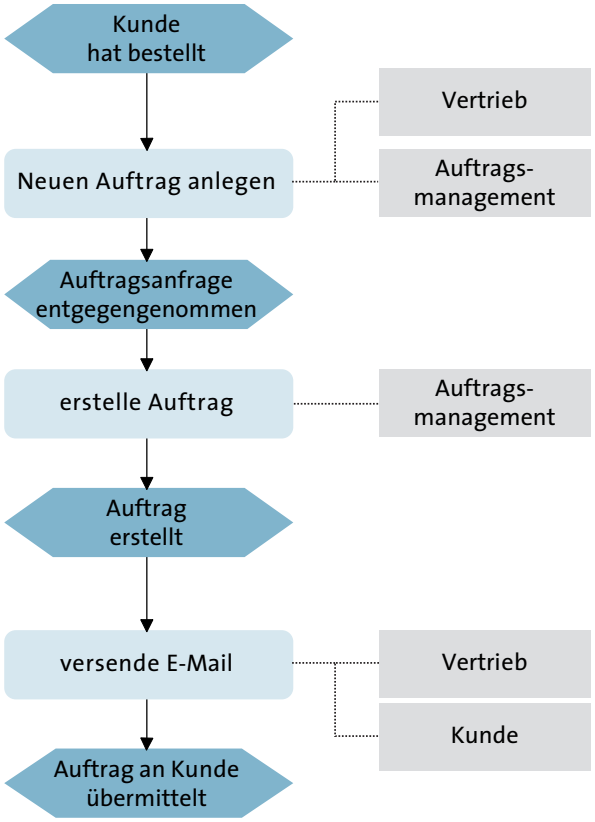


Abbildung 19: Prozessdifinition als Ereignisgesteuerte Prozesskette

was zu tun ist (z. B. organisationsübergreifend: Ereignis mit einer selbst initiierten internen Aktivität).

Architekturentwurf Schritt 1 – Identifzierung der Dienste

Als erstes stellt sich die Frage, welche Dienste benötigt werden und wie man sie aus den Geschäftsprozessen ableiten kann.

Hierfür gibt es keine pauschale Antwort. Dabei ist auch wichtig zu betrachten, ob der Schwerpunkt der SOA in einer unternehmensweiten oder der anwendungsinternen (Prozess-)Ebene liegt.

Bei der Betrachtung unternehmensweiter Prozesse sind die Akteure gute Dienstkandidaten, also z. B. ein Vertriebsdienst oder ein Auftragsmanagementdienst. Hier finden sich die Fachabteilungen wieder.

Anwendungsintern sind die fachlichen Objekte wie die Bestellung oder der Auftrag bzw. der Bestell- und Auftragsdienst gute Kandidaten.

Architekturentwurf Schritt 2 – Abbildung der Aktivitäten und Ereignisse

Die Abbildung der Aktivitäten und Ereignisse hängt sehr stark davon ab, wie der Prozess gesteuert wird und auch welche technischen Ansätze verwendet werden.

Bei der Prozess-Steuerung unterscheidet man die direkte Steuerung („Orchestrierung“) von der indirekten Steuerung („Choreographie“).

Bei der Orchestrierung wird die Prozessdefinition (z. B. von einem EPK nach BPEL überführt) in eine zentrale Steuerungskomponente geladen, die dann den Prozess verwaltet. Anhand der Prozessdefinition werden zur Laufzeit Prozessinstanzen erzeugt, deren Zustände durch die Prozess-Steuerung verwaltet werden. Ein neuer Zustand, also ein Ereignis ist damit das Ergebnis einer Aktivität eines Dienstes.

Beispiel: Nach der Definition des Bestellprozesses wird die Definition des Prozesses in eine Steuerungskomponente geladen. Zur Laufzeit wird eine neue Instanz eines Bestellprozesses erzeugt, wenn ein Kunde eine Bestellung auslöst. So wird von der Prozess-Steuerung die Aktivität „Neuen Auftrag Anlegen“ im Auftragsmanagementdienst ausgelöst, sobald ein Auftrag angelegt werden soll. Nach erfolgreicher Beendigung überführt die Steuerungskomponente die Instanz des Prozesses in den Zustand „Auftrag Angelegt“. Danach löst die Steuerungskomponente im Vertriebsdienst die Aktivität „Benachrichtige Kunde“ aus. Nach der Beendigung setzt die Steuerungskomponente den Zustand „Auftrag an Kunde übermittelt“.

Fazit: In diesem Szenario stellen Dienste typischerweise nur Aktivitäten in Form von Operationen bereit. Diese sollten zur Wiedererkennung den Namen der Aktivität aus der Prozessdefinition tragen. Es findet eine synchrone, Punkt-zu-Punkt Kommunikation mit der Prozess-Steuerung statt.

Bei dem Choreographie-Ansatz wird der Prozess indirekt gesteuert. Dienste rufen sich gegenseitig auf.

Hier empfiehlt sich, die Implementierung der Aktivitäten und Ereignisse von der Prozessdefinition, der Verwaltung der Prozessinstanzen und -variablen zu trennen. Letzteres kann die Steuerungskomponente z. B. über eine Rule Engine leisten.

Falls der Einsatz einer zentralen Prozesssteuerung nicht möglich ist (z. B. weil nicht geklärt werden kann, wer den Prozess steuern soll), ist es auch möglich, dass die Dienste ihren Teil der Steuerung selbst implementieren.

Schließlich müssen die Dienste nur wissen, welches Ereignis bzw. welche Aktivität ausgelöst werden muss, sobald die eigene Aktivität bzw. die Kommunikation des eigenen Ereignisses beendet ist. Auch parallele Ausführungen sind damit einfach zu realisieren.

Beispiel: Der Bestellprozess ist durch den Vertrieb gemeinsam mit dem Auftragsmanagement definiert worden. Der Vertrieb hat einen Vertriebsdienst und das Auftragsmanagent den Auftragsmanagementdienst implementieren lassen. Die Schnittstellen enthalten die vereinbarten Aktivitäten bzw. Ereignisse. Ein Kunde bestellt über das Vertriebsportal des Vertriebsmanagements. Das Portal

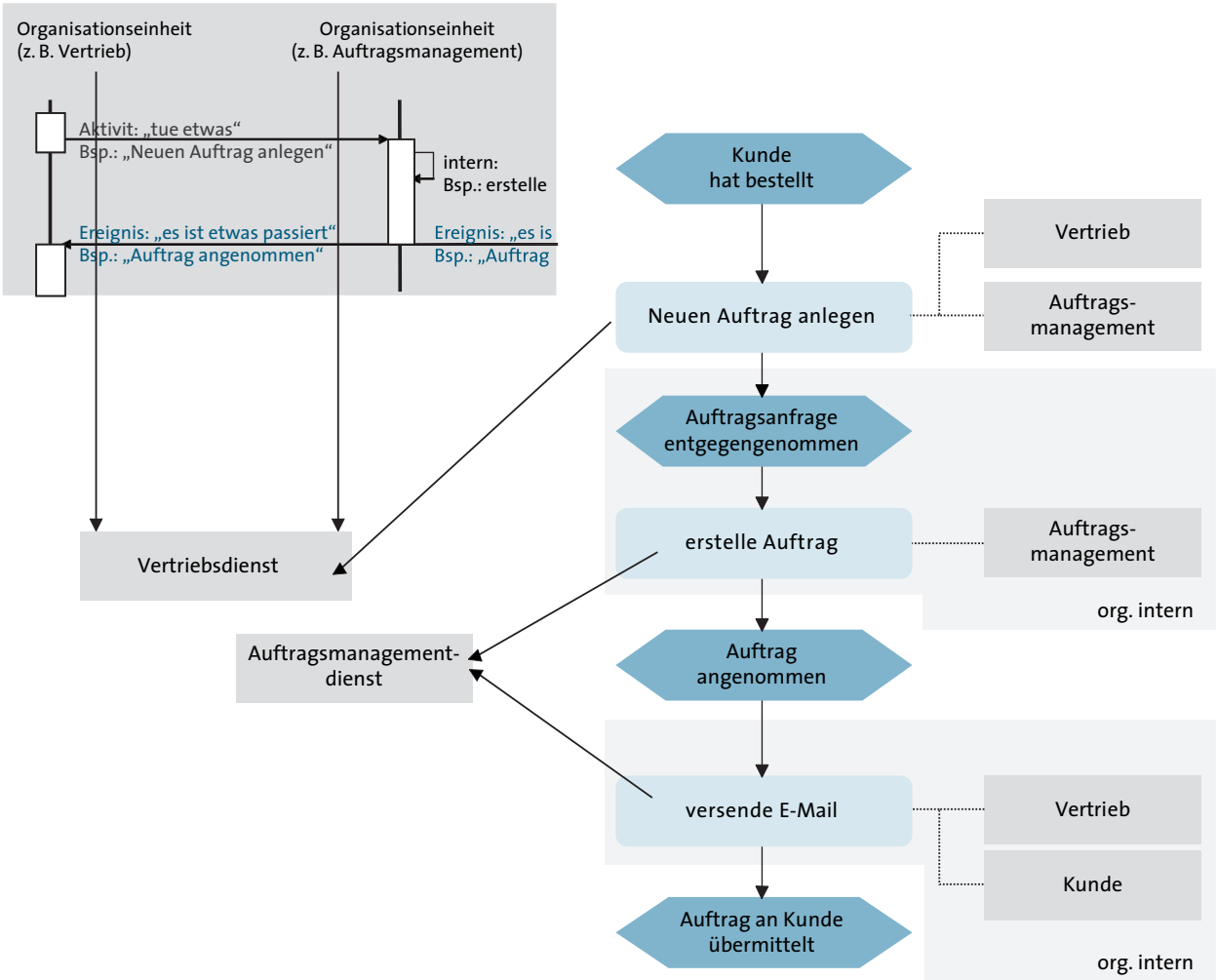


Abbildung 20: Identifizierung von Diensten anhand von Organisationseinheiten auf Unternehmensebene. Auch andere Kriterien sind denkbar. Zerlegung des EPKs in organisationsinterne und externe Prozesse für den besseren Vergleich

fordert den Auftragsmanagementdienst über die Aktivität „Neuen Auftrag anlegen“ auf, den Auftrag anzulegen. Mit der Anlage des Auftrags publiziert der Auftragsmanagementdienst das Ereignis „Auftrag angenommen“. Dieses Ereignis erhält u. a. der Vertriebsdienst. Er reagiert, indem er eine E-Mail generiert und dem Kunden zustellt. Danach kommuniziert der Vertriebsdienst das Ereignis „Auftrag an Kunden übermittelt“.

Es wird deutlich, dass der Choreographie-Ansatz Ereignisorientierung unterstützt, die man aus den ereignisorientierten bzw. nachrichtenorientierten Architekturen (MOMs, EDA) kennt. Er unterstützt sowohl eine 1-zu-1 als auch eine Publish-/Subscribe-Kommunikation. Der Ansatz bietet sich vor allem dort an, wo es z. B. aus organisatorischen Gründen nicht möglich ist, eine zentrale Prozess-Steuerung zu etablieren (wem gehört sie?) oder wenn die fachliche Abläufe durch die Verwendung von Ereignissen „loser gekoppelt“ werden sollen. Beispielsweise kann sich bei der Definition des Prozesses auf Unternehmensebene auf die Kommunikation zwischen den Organisationseinheiten konzentriert werden, bei denen internen Teilprozesse durch externe Ereignisse ausgelöst werden, die nicht extern abgestimmt bzw. publiziert werden müssen.

Dabei muss jedoch nicht auf eine „engere Kopplung“ durch Aktivitäten („Tue etwas“) verzichtet werden. Beides ist gemeinsam einsetzbar.

Während der Choreographie-Ansatz flexibler und organisatorisch einfacher zu etablieren ist, ist er in der SOA-Welt noch relativ neu. Die dazu gehörenden Standards sind noch nicht „marktreif“ bzw. akzeptiert. Der Orchestrierungs-Ansatz repräsentiert den momentanen Stand der Technik.

Einsatzgebiete von SOA-Architekturen

Je nach Einsatzgebiet ergeben sich unterschiedliche Ziele, die man mit der SOA erreichen will. Die Diskussionen ergeben sich häufig aus den unterschiedlichen Rollen der Beteiligten. Beispielsweise hat der Anwendungsarchitekt eine ganz andere Zielsetzung als der Unternehmensarchitekt. Deshalb ist es besonders wichtig, die Einsatzgebiete und auch die Architekturen zu gliedern. Eine SOA-Architektur für eine einzelne Anwendung kann ganz anders aussehen als eine SOA-Architektur für eine Anwendungslandschaft in einem Unternehmen.

Folgende Unterscheidung empfiehlt sich:

Einsatzgebiet	Mögliche Ziele
SOA auf Marktebene (Business to Business)	■ Leistungen durch Dienste anderer Firmen und Partner (z. B. Lieferanten) nutzen ■ an deren Prozessen teilnehmen bzw. in diese integrieren (Lieferketten) ■ Eigene Leistungen durch Dienste anbieten ■ Eigene Prozesse öffnen, damit sich Partner in diese integrieren können (Optimierung von Lieferketten/Lieferanten)
SOA auf Unternehmensebene (Application to Application)	■ Geschäftsnahe Kernfunktionalität der einzelnen Anwendungen in Form von Diensten als Beitrag zur Unternehmung bereitstellen ■ Auflösung von Redundanzen zwischen einzelnen Anwendungen durch Bestimmung der Kernfunktionalität von Anwendungen auf Dienstebene (Konzentration Kernfunktionalität) ■ Direkte, einfache, schnelle und kostengünstige Abbildung agiler Geschäftsprozesse auf Anwendungen ■ Eigene, interne Leistungen durch Dienste anbieten ■ Interne Leistungen durch die Integration anderer Dienste nutzen ■ Kostenreduktion und Agilität durch einheitliche Schnittstellentechnologien ■ Generelle Einbindung/Integration von Standardprodukten (Standard-Dienste) und Legacy-Anwendungen

Einsatzgebiet	Mögliche Ziele
SOA auf Komponentenebene (anwendungsintern)	■ Entkoppelung von Komponenten und Modulen ■ Einfache Nutzung unternehmensinterner Dienste ■ Extraktion und plattformunabhängige Wiederverwendung technischer Dienste ■ Nutzung von Standardkomponenten (z. B. Prozesssteuerung) ■ Integration einzelner (kleiner) Funktionalitäten einer bestehenden, proprietären Anwendung (z. B. Legacy-System)

Tabelle 1: Unterschiedliche Ziele für unterschiedliche Einsatzgebiete einer SOA

Daraus lässt sich folgende Empfehlung darüber ableiten, welche SOA-Komponenten in der Referenzarchitektur in dem entsprechenden Szenario verwendet werden sollten:

Ebene	Marktebene	Unternehmens-ebene	Komponentenebene (anwendungsintern)
Dienste	muss	muss	muss
Verzeichnisdienst (Repository)	empfohlen	muss	nur bei sehr vielen Komponenten
Service Bus	nein	empfohlen	nur bei sehr vielen Komponenten
Prozessteuerung	nein	empfohlen	wenn stark ablaufgeprägt
Dienst-Monitor	empfohlen	empfohlen	empfohlen

Tabelle 2: Empfohlener Einsatz von SOA-Komponenten nach Einsatzgebiet

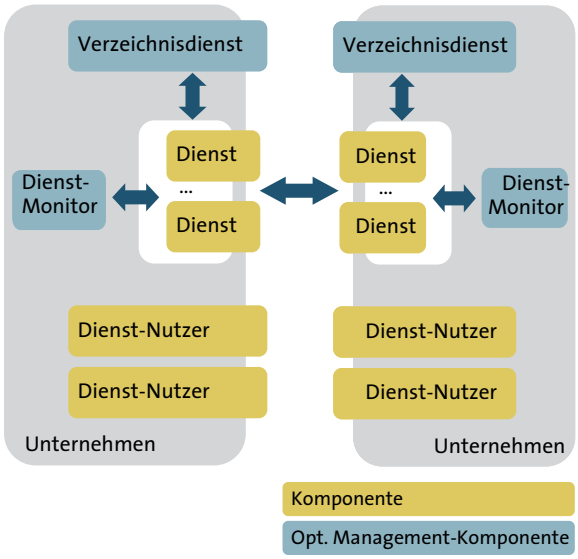


Abbildung 21: Empfohlene Komponenten für eine SOA auf Marktebene (Business to Business)

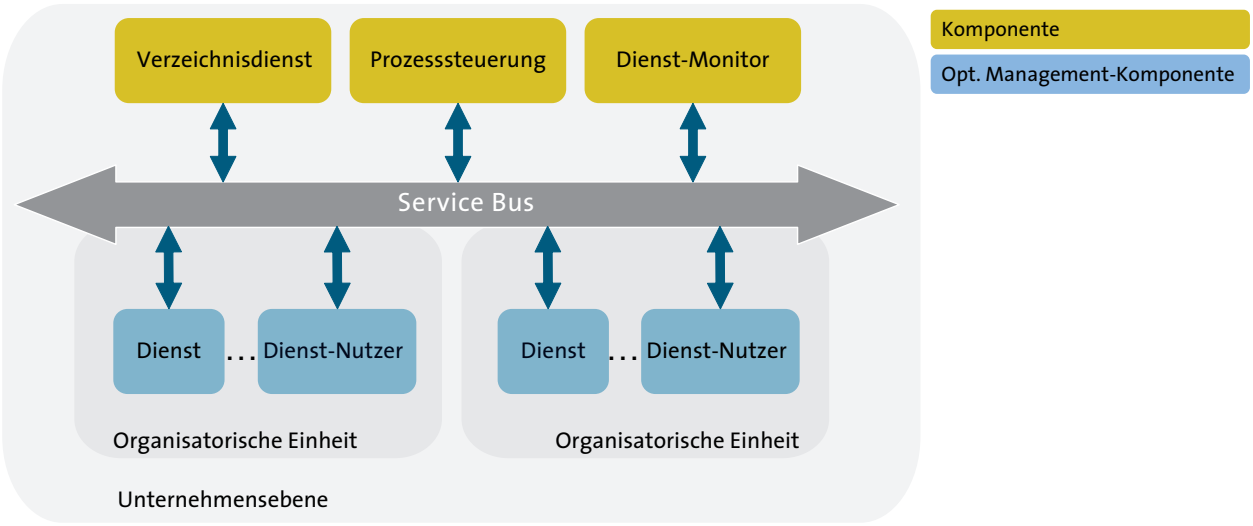


Abbildung 22: Empfohlene Komponenten für eine SOA auf Unternehmensebene (Application to Application)

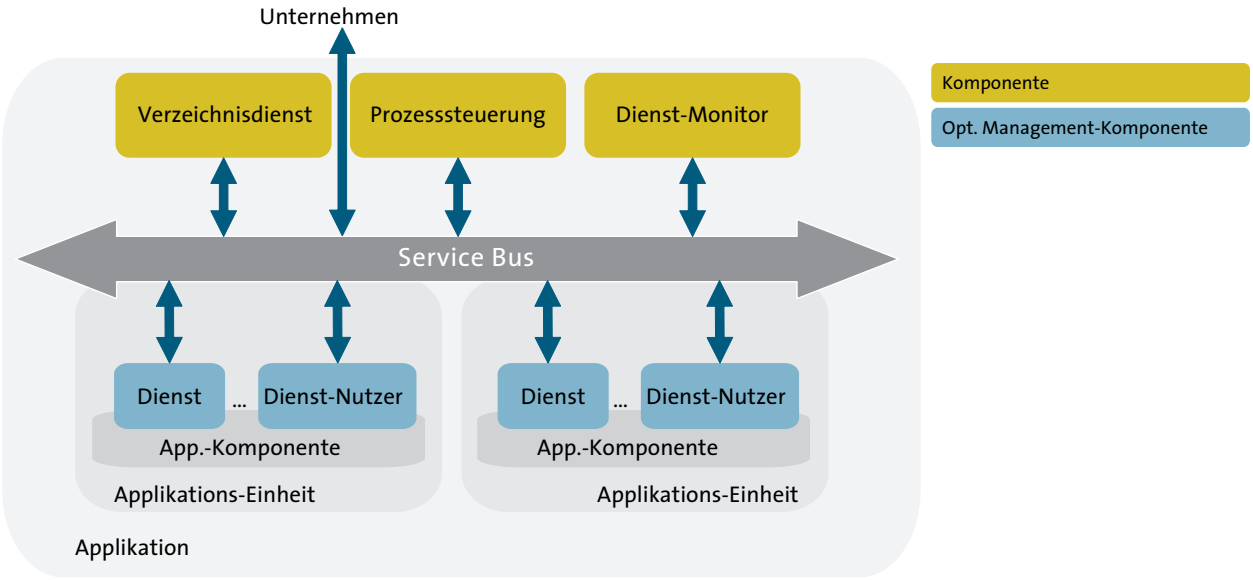


Abbildung 23: Empfohlene Komponenten für eine SOA auf Applikationsebene (hier voll ausgebaut mit internem Service Bus)

Warum sollte man welche Komponente einsetzen:

Komponente	Einsatzgebiet	Begründung der Empfehlung
Dienste	alle Ebenen	■ Aus dem Paradigma ergibt sich, dass immer Dienste benötigt werden.
Verzeichnisdienst	Marktebene	■ Transparente Gestaltung von Diensten und Leistungen ■ Einfache Bereitstellung mit einheitlichem Sicherheitsansatz möglich. ■ Wichtig für die Einführung: Akzeptanz bei den Partnern
	Unternehmens-ebene	Governance und Transparenz: ■ Vermeidung von „Wildwuchs“ bei Diensten ■ Kontrollierter Aufbau und Betrieb der Dienstlandschaft ■ Dokumentation der bereitgestellten Dienste
	Anwendungs-ebene	Kontrolle und Transparenz bei Einsatz vieler internen Dienste: ■ Vermeidung von „Wildwuchs“ ■ Kontrollierter Aufbau und Betrieb der Architektur ■ Dokumentation
Service Bus	Marktebene	■ Empfiehlt sich i. d. R. nicht in der Anwendung zwischen Unternehmen, da es hier häufig zu einer Punkt-zu-Punkt Kommunikation kommt. ■ Ausnahmen wie z. B. bei SWIFTNet bestätigen die Regel.
	Unternehmens-ebene	■ Reduzierung der Anzahl der Verbindungen zwischen Systemen ■ Standardisierte Integration bestehender Anwendung ■ Standardisierung der Kommunikation ■ Vereinfachung des Managements und Verbesserung der interne Kommunikation ■ Je nach Größe empfiehlt sich der Aufbau von Bus-Hierarchien.
	Anwendungs-ebene	Bei Einsatz vieler interner Dienste: ■ Reduzierung der Anzahl der Verbindungen zwischen Systemen und Komponenten ■ Standardisierte Integration bestehender Anwendung ■ Standardisierung der Kommunikation
Prozesssteuerung	Marktebene	■ Schwierig, Prozesssteuerungen auf Marktebene einzusetzen: Wer besitzt bzw. konfiguriert den Prozess? ■ Der Verzicht ist deshalb sinnvoll.
	Unternehmens-ebene	■ Dokumentation und Transparenz der Geschäftsprozesse und deren technische Abbildung ■ Bessere Geschäftsprozessüberwachung (Monitoring) ■ Fachliche Nähe insbesondere bei Problemen im Betrieb ■ Trennung von Prozess und Prozesselementen
	Anwendungs-ebene	■ Wie auf Unternehmensebene ■ Einsatz sehr sinnvoll, wenn die Anwendung sehr stark ablaufgeprägt ist
Dienst-Monitor	alle Ebenen	■ Auf allen Ebenen empfohlen ■ Besonders zwischen Unternehmen, da die Benutzung der Dienste (Leistung) bzw. die Fachseite i. d. R. vertraglich geregelt sind. Ein Ausfall hätte finanzielle Auswirkungen und zieht ggf. Strafen nach sich

Weitere wichtige Empfehlungen zum Entwurf einer SOA

Festlegung von Namenskonventionen

Für die Sicherung der einheitlichen Abbildung von Geschäftsprozessen ist es unumgänglich, Namenskonventionen für Aktivitäten und Ereignisse zu definieren. Das hilft besonders, den Unterschied zwischen Aktivität und Ereignis deutlich bzw. erkennbar zu machen.

Das könnte beispielsweise so aussehen:

Aktivität: Substantiv + Verb (aktive Aufforderung, dass der Dienst etwas tut), Bsp.: „Neuen Auftrag anlegen“

Ereignis: Substantiv + Adverb (der Dienst signalisiert, dass etwas getan worden ist. Der Empfänger kann daraufhin entscheiden, ob und welche Aktivität (intern) ausgelöst wird), Bsp.: „Auftrag angenommen“

Vermeidung von Diensthierarchien innerhalb einer SOA

Die Hierarchisierung ist ein generell ein gutes Mittel zur Strukturierung. In Service-orientierten Architekturen sind Hierarchien allerdings insbesondere bei der Orchestrierung ein Problem, da durch die Einführung einer Hierarchie ein Dienst den untergeordneten Dienst aufruft und damit die Prozesssteuerung umgeht. Eine Hierarchie deutet darauf hin, dass es verschiedene Architekturebenen adressiert werden, z. B. ein Dienst auf Unternehmensebene steht in Verbindung mit einem Dienst auf Anwendungsebene. Es ist wahrscheinlich, dass es sich hier auch bei der Prozessgestaltung um unternehmensweiten Teilprozess mit einem organisationsinternen Ablauf handelt.

Die Abbildung von Geschäftsprozessen auf Web-Services

Jeder Dienst kann auf einen W3C Web-Service abgebildet werden. Dabei ist wichtig, zu differenzieren, dass ein W3C Web-Service nicht notwendigerweise http als Transportprotokoll verwenden muss, sondern z. B. auch MQ.

Wie sollten nun Prozesselemente idealerweise auf Web-Services abgebildet werden?
Dabei sind folgende Szenarien zu unterscheiden:

- Eine Prozesssteuerung wird eingesetzt (Orchestrierung)
- Keine direkte Prozesssteuerung und ein ereignisfähiges Transportprotokoll wird eingesetzt
- Keine direkte Prozesssteuerung und kein ereignisfähiges Transportprotokoll wird eingesetzt
- In allen drei Fällen werden Aktivitäten zu gleichnamigen Web-Service-Operationen, die zuerst als Request eine Input- und als Response eine Output-Nachricht verwenden (wie eine Methode in einem Objekt).

Beispiel: Aus „Neuen Auftrag anlegen“ aus dem Auftragsmanagement wird im Web-Service eine Operation neuenAuftragAnlegen mit einer Input und einer nachfolgenden Output Message.

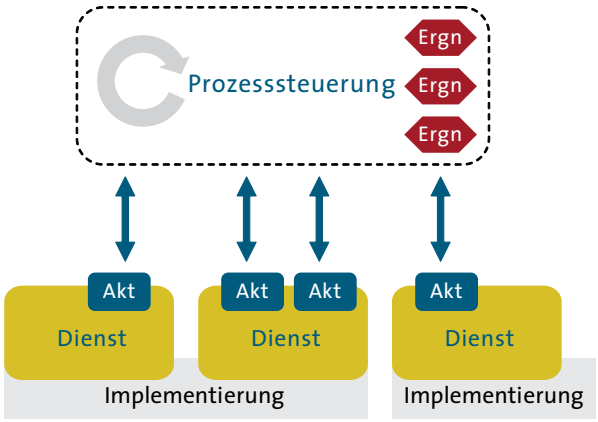


Abbildung 24: Die Standardlösung (Orchestrierung mit Prozessteuerung)

Die Abbildung von Ereignissen ist abhängig von den oben beschriebenen Szenarien. Wird eine Prozesssteuerung (Orchestrierung) eingesetzt, so werden die Ereignisse (wie bereits eingangs beschrieben) intern behandelt und werden nicht Bestandteil der Dienstschnittstelle.

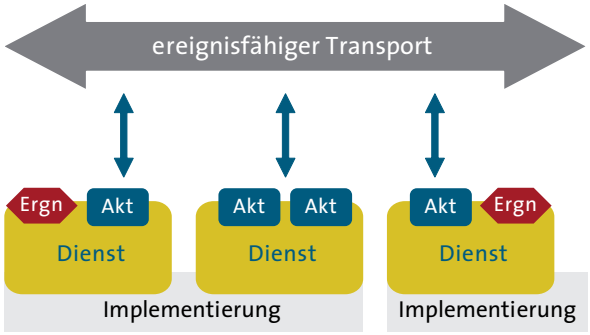


Abbildung 25: Die ereignisfähige Lösung ohne direkte Prozessteuerung (Choreographie mit ereignisfähigem Transport)

Werden keine direkte Prozesssteuerung und ein ereignisfähiges Transportprotokoll eingesetzt (Szenario b), werden Ereignisse zu gleichnamigen Web-Service-Operationen, die eine Output-Nachricht versenden. Wird eine Messaging-Infrastruktur verwendet, kann diese Nachricht auch verteilt werden (Publish-/Subscribe).

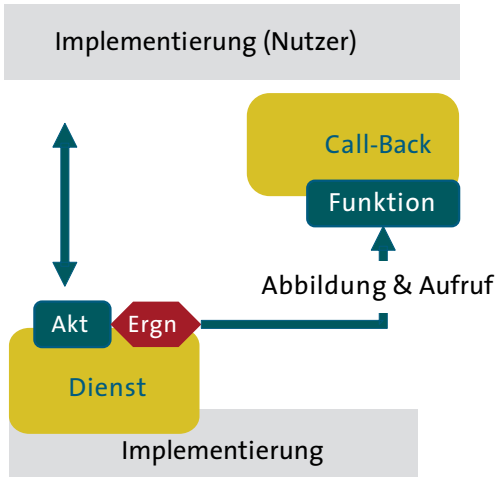


Abbildung 26: Die ereignisfähige Lösung ohne direkte Prozessteuerung mit einem Workround (Choreographie ohne ereignisfähigem Transport, z. B. http)

Bei Szenario c) muss man sich eines Workarounds bedienen, da Ereignisse nicht aus der Dienstschnittstelle heraus versendet werden können. Das Ereignis als Aktivität in einen fremden Dienst zu platzieren, würde der Fachlichkeit nicht entsprechen und falsche Abhängigkeiten zwischen den darunter liegenden Implementierungen schaffen. In diesem Fall müssten alle Empfänger eine Call-Back-Schnittstelle bereitstellen, die der Dienstbesitzer definiert, aber der Dienstnutzer bereitstellt. Die Implementierung des Dienstes ruft dann die Dienstschnittstelle des „Call-Back Dienstes“ auf.

Was ist und was wird möglich?

Wo die SOA-Entwicklung momentan steht und wo sie sich hin entwickelt, lässt sich sehr gut anhand der zuvor beschriebenen drei Szenarien darstellen. Der gängigste Ansatz heutiger SOA-Lösungen ist die der Orchestrierung. Dieser setzt jedoch einen starken Organisationsgrad voraus, was sich in der unternehmensweiten Einführung als sehr schwierig darstellt. Wer gibt schon gerne die Kontrolle des Prozesses an den Besitzer eine Prozesssteuerung ab. Andererseits zwingt dieser Ansatz Unternehmen auch in der IT eindeutig die Prozessbesitzer zu definieren. In der Praxis ist das allerdings sehr schwierig durchsetzbar.

Ereignisorientierung mit einer entsprechenden Choreographie ermöglicht eine Entkoppelung von Geschäftsprozessen und eine 1:n-Kommunikation (Publish-/Subscribe). Nicht umsonst sind EDA (Event Driven Architectures) auf MOMs (Message oriented Middleware) sehr beliebt und auf Unternehmensebene ein weitverbreiteter Architekturansatz. Der Ansatz ist wesentlich flexibler, bietet Möglichkeiten, die Geschäftsprozesse wesentlich direkter abzubilden und ist sehr viel einfacher zu organisieren und durchzusetzen. Das sich SOA in diese Richtung entwickelt (also mit dem EDA zusammenwächst) ist ziemlich deutlich an der Tatsache zu erkennen, dass entsprechenden Standards wie WS-Eventing, WS-Notification, WS-Reliable-Massaging und WS-Choreography entwickelt werden. Sie sind allerdings noch nicht etabliert. In der Zwischenzeit erlauben SOAP-fähige ESB-Produkte erste kleine Schritte in die Richtung EDA. Im Hinblick auf zukünftige Entwicklungen ist es wichtig, beide Paradigmen zu verstehen und

im Entwurf einer komplexen Architektur zu berücksichtigen. Demnach ist EDA nicht einfach eine Weiterentwicklung von SOA, sondern eine Komplettierung im Hinblick auf erforderliche Architekturkonzepte.

4.3 Security

SOA-Security – Mehr als ein technisches Thema

Das Thema „Security" hat sich in den letzten Jahren – nicht zuletzt durch zahlreiche spektakuläre Zwischenfälle – von einem technischen Spezialgebiet für Insider zu einem geschäftsrelevanten Faktor entwickelt. Dies gilt in ganz besonderem Maße für eine Service Oriented Architecture, verfolgt diese doch den Ansatz, modulare und neue Anwendungselemente zu neuen Lösungen zu verknüpfen. So entsteht eine offene Umgebung, deren Nutzung durch geeignete Techniken kontrolliert werden muss. Art und Umfang dieser Kontrolle werden durch geschäftliche Vorgaben bestimmt. Im Folgenden wird eine flexible Architektur vorgeschlagen, mit der solche Vorgaben umgesetzt werden können.

Dabei sind die Anforderungen an die verschiedenen Aspekte bzw. Disziplinen von Security die gleichen wie in jeder anderen IT-Umgebung: Identitäten und Berechtigungen müssen verwaltet werden, Benutzer werden authentisiert, Zugriffsberechtigungen werden überprüft, Zugriffe und Zugriffsversuche werden im Log aufgezeichnet, mit kryptografischen Methoden werden Daten signiert (Integrität) oder verschlüsselt (Vertraulichkeit). Das Ganze unterliegt einer Sicherheitsrichtlinie (Policy), deren Einhaltung (Compliance) überwacht und permanent sichergestellt werden muss. Da es sich bei SOA typischerweise um das Zusammenspiel verteilter Anwendungselemente handelt, kommen folgende Aufgaben hinzu: Die häufige Umwandlung von Identitäten, die Verwendung von Industriestandards sowie die Wiederverwendbarkeit der Security-Dienste selbst.

Gerade der letzte Punkt erfordert eine Grundsatzentscheidung: Will man die Security-Funktionen als Service der Infrastruktur realisieren und somit die einzelnen Anwendungen von dieser Aufgabe weitgehend befreien, oder lassen sich bestimmte Dinge nur im Kontext der Anwendung realisieren? Die Vorteile einer Service-Infrastruktur liegen auf der Hand: Die Anwendungsentwickler sind von der Beschäftigung mit der – häufig doch recht komplexen – Security-Thematik entlastet und können sich ganz auf die Business-Logik konzentrieren. Auch müssen sie das Rad nicht immer wieder neu erfinden.

Eine Referenzarchitektur für SOA-Security

Neben den Kern-Services enthält das SOA-Architektur-Referenzmodell die unterstützenden Komponenten. Uns interessiert in diesem Zusammenhang speziell die Komponente „IT Service Management". Sie enthält zahlreiche Überwachungs- und Managementkomponenten – auch diejenigen, die zur Security gehören. Hier knüpft die IBM SOA-Security-Referenzarchitektur an, wie in der folgenden Grafik dargestellt wird. Sie besteht aus drei logischen Blöcken.

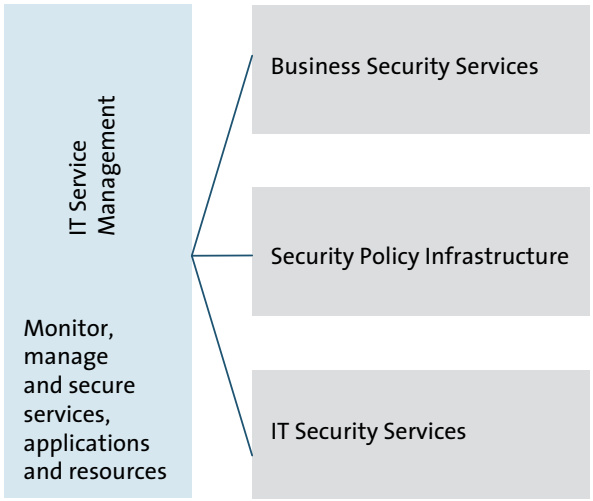


Abbildung 27: SOA-Security-Referenzarchitektur

Der Block „IT Security Services" steht für den Ansatz, die Security selbst als Service zu realisieren. Ein eigener Block für „Business Security Services" unterstreicht den engen Bezug der Security zu den Geschäftszielen einer SOA: Hier wird die Security Policy direkt aus der übergeordneten

Business-Policy abgeleitet. Es wird angestrebt, dies weitgehend zu automatisieren. Dazu dient die mittlere Schicht „Security Policy Infrastructure“. Die drei Funktionsblöcke werden im Folgenden weiter aufgeschlüsselt und erläutert.

Identity Services	Authorization & Privacy Services	Audit Services
Authentication Services	Confidentiality & Integrity Services	Non-repudiation Services

Abbildung 28: IT Security Services.

Zentrale Security Services können von verschiedenen Komponenten genutzt werden. Neben den Endbenutzern und den Dienstanbietern sind dies beispielsweise Infrastrukturkomponenten wie Gateways, Proxies, Anwendungsserver, Datenbankserver, Betriebssysteme. Durch Mehrfachverwendung der Dienste erreicht man Konsistenz und Einsparung von Kosten. Die Liste der Services selbst entspricht den klassischen Disziplinen, wichtig sind die Schnittstellen und der Abstraktionsgrad. Identity Services umfassen Benutzerverzeichnisse sowie gegebenenfalls deren Abgleich (falls es mehrere davon gibt), die Provisionierung von Benutzerdaten verbunden mit der dazugehörenden Identity-Management-Funktionalität sowie – besonders verbreitet im SOA-Umfeld – die Unterstützung voneinander unabhängiger Systeme, sei es im gleichen Unternehmen oder unternehmensübergreifend (Identity Federation).

Authentication Services sollen unterschiedliche Verfahren (z. B. Passwort, Zertifikat, Token, Biometrie) und Protokolle (z. B. Kerberos) bereitstellen, mit denen sich ein Benutzer ausweisen kann. Im SOA-Umfeld (speziell bei Web Services) kann dies sowohl bei einem Service Consumer als auch bei einem Service-Provider geschehen. Auch eine Umwandlung einer einmal festgestellten Identität auf ein anderes Format kommt dann häufig vor.

Authorization and Privacy Services stellen die Entscheidung bereit, ob ein Zugriff eines Benutzers auf eine bestimmte Ressource erlaubt ist. Die Entscheidung basiert auf einer Policy und nimmt Bezug auf die Identität des Benutzers und dessen relevante Attribute. Die Aufgaben sind verteilt zwischen Policy Decision Point (PDP – Entscheidungsfindung) und Policy Enforcement Point (PEP – Entscheidungsdurchsetzung).

Confidentiality and Integrity Services schützen die eigentlichen Nachrichten bzw. Teile (Elemente) davon vor Einsichtnahme (Confidentiality), Manipulation (Integrity), Fälschung des Absenders und Wiederholung einer Nachricht. Dies kann auf Basis des Transportkanals geschehen (z. B. SSL – nicht typisch für SOA) oder auf Basis der einzelnen Nachrichtenpakete bzw. ausgewählter Elemente.

Non-repudiation Services stellen für beide beteiligten Seiten nachweisbar sicher, dass eine Transaktion wie oben beschrieben stattgefunden hat. Diese Funktionalität wird auf Basis kryptographischer Verfahren ausgeführt.

Audit Services bieten eine zentrale Infrastruktur zum Erstellen, Speichern und Verwalten von relevanten Ereignissen. Relevante Ereignisse können z. B. Anmeldeversuche, Verletzungen einer Policy, Ausfälle von Security Ser-

Policy Administration	Policy Distribution & Transformation	Policy Decision & Enforcement	Monitoring & Reporting

Abbildung 29: Security Policy Infrastruktur

vern oder Zugriffe auf sensitive Daten sein. Eine Auditing Policy legt das Verhalten dieser Dienste fest.

Unabdingbar für die Wiederverwendbarkeit der hier genannten Services ist die Benutzung geeigneter Standards. Als Beispiel sei genannt der Bereich Web Services Security.

Policies im Kontext einer SOA stehen in einer Top-down-Struktur, deren oberste Ebene die Business Policies sind. Diese werden zu servicespezifischen Policies verfeinert, aus denen wiederum die Anforderungen an das Verhalten der Infrastruktur abgeleitet werden. Soweit möglich sollten diese Verfeinerungen automatisiert werden. Wenn eine zentrale Service Registry vorhanden ist, kann diese als Speicher für die Policies verwendet werden. Von dort aus erfolgt die Weiterverteilung. Wichtig ist, dass die jeweils gültigen Policies feststellbar sind, Änderungen protokolliert werden und somit für jeden Zeitpunkt nachvollzogen werden kann, welche Policies aktiv waren.

Unter Policy-Administration verstehen wir in diesem Zusammenhang das Erzeugen, Löschen, Verwalten und Importieren/Exportieren der gerade beschriebenen Policies mit den jeweils verfügbaren Tools. Meist ist diese Funktionalität Bestandteil der Komponenten, die auch den zugehörigen Service selbst realisieren. So wird etwa eine Access Policy mit der Managementkomponente des verwendeten Access-Control-Produktes bearbeitet. Dies geschieht in der Regel mit Hilfe von einer oder mehreren der drei Varianten: Web Interface, Command Line Interface (CLI) oder Programmierschnittstelle (API). Die beiden letzteren eignen sich für eine Automatisierung durch Skripte oder eigene Programmierung. Handelt es sich dabei um ein herstellerspezifisches API-Format, ist eigene Programmierung der einzige Weg für eine Automatisierung. Bei standardisierten Formaten (z. B. XML-basierte Formate wie XACML) ist denkbar, dass der Markt Alternativen hervorbringt, so wie es heute etwa für LDAP-Verzeichnisse offene LDAP-Browser gibt.

Ein Policy Decision Point (PDP) verwendet die Policy-Daten, um eine Entscheidung darüber herbeizuführen, ob der gerade erfolgende Request zugelassen werden soll oder nicht. Diese Entscheidung wird einem Policy Enforcement Point (PEP) zur Verfügung gestellt, der die eigentlichen Zugriffe überwacht. Der PEP verfügt über die Mittel, die Zugriffe zu gestatten oder zu unterbinden. Durch diese Arbeitsteilung ist es möglich, mit einem zentralen PDP mehrere – auch verschiedene – spezialisierte PEPs zu betreiben, deren Mechanismen der Art des jeweilige Zugriffs angepasst sind (z. B. HTTP-Proxy für HTTP-Requests, Kernel-Exit für Unix-Systemaufrufe).

Policy Distribution and Transformation beschreibt, in welchen Formaten die zu verteilenden Policies formuliert und ggf. transformiert werden. Soweit möglich, sollten hier Standards wie WS-Policy, WS-Security Policy oder XACML verwendet werden. Die Verteilung der Policies muss mit abgesicherten Methoden erfolgen.

Abbildung 11: Business Security Services

Business Security Services

Unter Business Security Services verstehen wir in einer SOA-Umgebung die Prozesse und Verfahren, mit denen die Umgebung gesteuert und den Geschäftsbedürfnissen angepasst wird. Die eigentlichen Funktionalitäten werden von den Diensten des Funktionsblocks „IT Security Services“ erbracht. Die Business Security Services lassen sich grob in fünf Kategorien unterteilen.

„Governance, Risk and Compliance“ beschreibt die Kontrolle über die Integrität des Gesamtsystems, schätzt das Restrisiko durch Security-Verletzungen ab und misst die Übereinstimmung des Systems mit den Anforderungen der Business Policies, insbesondere mit denjenigen, die durch gesetzliche oder andere regulatorische Vorgaben zustande kommen. Dieses Thema ist nicht auf Security-Aspekte beschränkt, doch speziell hier kommt es darauf an, dass eine klare Verantwortungsstruktur abgebildet wird. Audits werden in diesem Kontext als wichtiges Instrument verwendet.

Das Trust Management behandelt sowohl die nicht technischen Aspekte (Verträge, Absprachen, DOU) wie auch die technischen Aspekte (Kryptoverfahren, Key-Management, Attributmanagement) des Vertrauensverhältnisses zwischen den beteiligten Organisationen bzw. Bereichen.

„Identity and Access“ regelt das Identity Lifecycle Management, welches typischerweise durch ein auf Rollen aufgebautes, von HR-Daten gespeistes Identitätsmanagement realisiert ist, welches um ein ressourcenbezogenes Management von Zugriffsrechten sowie Self-Service-Funktionen und Genehmigungsprozesse ergänzt wird.

„Data Protection and Disclosure Control“ legt die Policies fest, welche die bereits beschriebenen „Confidentiality and Integrity Services“ steuern. Insbesondere im Bereich der personenbezogenen Daten (Information Privacy) werden hier Schnittstellen nicht nur für den Administrator, sondern auch für den Endbenutzer benötigt. Der Endnutzer muss über die Verwendung seiner Daten entscheiden können. Ferner gilt es, solche Entscheidungen aufzuzeichnen und jederzeit (z. B. durch entsprechende Reports) nachvollziehbar zu machen.

Mit „Secure Systems and Networks“ meinen wir die Strategie im Bereich der „klassischen“ Security-Themen wie Firewalls, das Härten von Betriebssystemen gegen Angriffe, Intrusion Detection, Virenschutz sowie Security-spezifisches Patchmanagement (häufig eingebettet in allgemeines Patchmanagement).

Für alle genannten Punkte muss es Businessprozesse und Policies geben, nach denen die Steuerung erfolgt.

Fazit

Es wird empfohlen, bei SOA-Implementierungen die Security-Aspekte von Anfang an in allen Projektphasen einzubeziehen. Es gilt den häufig auftretenden Fehler zu vermeiden, Security erst nachträglich einer fast oder ganz fertigen Lösung aufzusetzen. Eine solche Vorgehensweise wäre gefährlich, denn eine Designänderung könnte erforderlich werden, um Sicherheitslücken zu schließen. Diese Änderung wird häufig aufgrund von Zeitdruck weggelassen oder nicht ausreichend getestet. So entstehen unsichere Systeme, ganz zu schweigen von den unnötig hohen Kosten.

Leitfaden zu einer sicheren SOA

SOA ohne Security ist undenkbar. Neben den „klassischen“, bei jeder IT-Lösung vorhandenen Aufgaben gibt es auch spezielle, durch den verteilten Ansatz von SOA bedingte Anforderungen. Es empfiehlt sich, eine in das Gesamtmodell eingebettete Architektur zugrunde zu legen. Folgende Punkte gilt es zu beachten:

Eine Grundsatzentscheidung ist zu treffen, welche Security-Funktionen ihrerseits als Services der Infrastruktur realisiert werden sollen.

Die klassischen Aufgaben der Security müssen nicht neu erfunden werden, sie bekommen allenfalls SOA-spezifische Ausprägungen: Identitäten verwalten,

Authentisierung, Autorisierung, Auditing, Kryptographie, Policies, Compliance.

Bedingt durch die Heterogenität einer SOA entstehen zusätzliche Anforderungen an die Umwandlung von Identitäten sowie die Herstellung von Vertrauensbeziehungen (Trust).

Die Einbettung in die Gesamtarchitektur sollte im Rahmen von IT-Servicemanagement geschehen.

Die Aufgaben der SOA Security zerfallen in die drei Bereiche Security Management, Security-Policy-Infrastruktur und die eigentlichen Security Services.

Die Security Services fassen die genannten Aufgaben zu Funktionsblöcken zusammen: Identity, Authentication, Authorization/Privacy, Data/Message Protection, Auditing.

Die Policy-Infrastruktur ist verantwortlich für die Administration der Policies (Regeln), ihre Ableitung aus den geschäftlichen Vorgaben, ihre Verteilung an die Laufzeitumgebung, die Herbeiführung regelbasierter Entscheidungen (Policy Decision) sowie deren Umsetzung (Policy Enforcement). Die Regeln sind Bestandteil der Servicebeschreibungen. Sie beschreiben, wie mit den Services kommuniziert werden muss und welche Randbedingungen dabei einzuhalten sind. Sind verschiedene Dienstean-bieter beteiligt, wird auch deren Zusammenspiel definiert.

Security Management beinhaltet die Tools und Schnittstellen, die zur Bereitstellung und Verwaltung einer sicheren und stabilen Laufzeitumgebung zur Verfügung stehen.

Für die Erfolgskontrolle einer SOA-Lösung kommt dabei dem Auditing eine besondere Rolle zu. Es unterstützt u. a. Security-Audits, in denen überprüft wird, ob die Implementierung des Systems korrekt ist. Unabdingbar für das Zusammenspiel der Komponenten einer solch komplexen Umgebung, wie sie eine SOA nun einmal ist, ist die Verwendung von standardisierten Formaten und Protokollen, für Security sind dies beispielsweise WS-Trust, SAML, Kerberos, XACML, CBE, PKI.

Tiefergehende Informationen zur Security-Thematik finden Sie in Kapitel 7 „SOA und Security“ (Seite 104)

Weiterführende Links:

Understanding SOA Security Design and Implementation
http://www.redbooks.ibm.com/abstracts/sg247310.html

Oasis Standard
http://www.oasis-open.org/home/index.php

Web Services Security Roadmap
http://www-128.ibm.com/developerworks/library/specification/ws-secmap

5 Umsetzung und Betrieb

5.1 SOA-spezifisches Vorgehen

Der Übergang von monolithischen Anwendungen zu strukturierten Services einer Service-orientierten Architektur führt auch zu Veränderungen der betrieblichen Abläufe und Organisation. Die Veränderungen sind durch die Trennung von Dienstrealisierung und Dienstplattform und durch die lose Kopplung der Services zur Abbildung der Geschäftsprozesse eines Unternehmens bedingt. Da im Extremfall erst zur Laufzeit die aktive Nutzung von Services ermittelt wird, ändern sich die Ressourcenanforderungen dynamisch, was sich auch auf die betrieblichen Prozesse auswirkt. Das Management der IT-Services muss sich an diesen Anforderungen neu ausrichten und entsprechende Veränderungen der IT-Organisation und deren Prozessen herbeiführen. Diese Veränderungen ermöglichen neue Chancen zur Ermittlung des Wertbeitrages der IT für die Geschäftsprozesse im Unternehmen.

Veränderungen der IT-Prozesse

Generelle Unterschiede in der Betriebsführung

Generell unterscheidet sich durch die Zerlegung der Geschäftsfunktionalität in einzelne Dienste die Betriebsführung von SOA-basierten Systemen von der konventioneller Client/Server-Architekturen: Neben der SOA-Infrastruktur müssen auch die einzelnen Dienste im Rahmen des SOA-Governance-Prozesses überwacht werden. Dieser muss sicherstellen, dass notwendige Änderungen an einzelnen Diensten so erfolgen, dass auch bestehende Anwendungen den Dienst weiterhin nutzen können (Ein Dienst besteht konzeptionell aus einer Schnittstelle, die die Funktionalität des Dienstes beschreibt, und der eigentlichen Implementierung. Bei konstanter Schnittstelle kann die Implementierung für den Dienstnutzer transparent geändert werden. Damit können auf einer SOA basierende Anwendungen schnell instabil werden, ohne dass die Ursache dafür identifiziert werden kann. Das Problem kann nur organisatorisch im Rahmen eines SOA-Governance-Prozesses gelöst werden.).

Aus der Überwachung weniger aber komplexer Fachanwendungen wird eine Überwachung einer Vielzahl von Diensten, was allein auf Grund der Anzahl zu einer Herausforderung werden kann. Bei der Abschätzung der daraus erwachsenen Folgen muss berücksichtigt werden, dass die Überwachung eines einzelnen Dienstes wesentlich einfacher ist als die einer komplexen Anwendung. Außerdem bieten viele Hersteller schon heute Werkzeuge an, die helfen, das oben genannte Mengenproblem zu beherrschen.

Prozesse beim Dienst-Nutzer

Services werden von Service-Providern zur Verfügung gestellt. Der Service-Provider kann eine Leistung der eigenen IT-Organisation oder die einer externen Organisation sein. Um Verfügbarkeit, Qualität und Leistungsfähigkeit des Service durch den Service-Provider sicherzustellen bedarf es abgestimmter und vereinbarter Regeln und Richtlinien auf deren Basis die Lieferung und Nutzung des Service erfolgt: Neben einem technischen Kontrakt, der die technologische Möglichkeit des Zugriffs auf einen Dienst regelt, existiert ein logischer Kontrakt, das Service-Level-Agreement (SLA). Ein SLA regelt die Rahmenbedingungen für die Nutzung von Diensten. Im Rahmen eines SLA muss z. B. für jeden Dienst festgelegt werden, wann und wie ein Dienstnutzer in die Service-Management-Prozesse (s. u.) des Dienstanbieters einbezogen wird. Darüber hinaus wird festgelegt, welche Möglichkeiten ein Dienstnutzer erhält, seine Services in einer definierten Testumgebung zu testen. Es ist Aufgabe des Service-Level-Managements beim Dienstnutzer, die SLAs zu überwachen und Änderungen in den eigenen Change- und Release-Management einzubeziehen. Dies ist aktuell eine der größten Herausforderungen, weil das Geflecht der Abhängigkeiten mit steigender Zahl von Services und Servicenutzern bei ungeeigneten Prozessen und mangelnder Dokumentation

schnell unübersichtlich werden kann.
Ohne entsprechende Werkzeuge zur Verwaltung von Service-Versionen und der den jeweiligen Nutzern zugesicherten SLAs ist diese Teilaufgaben des Release-Management nur schwer zu meistern. Dies gilt insbesondere dann, wenn verschiedene produktive Versionen eines Service koexistieren. Normale Release-Wechsel und Patches zur Fehlerbehebung sind folglich mit einer eingehenden Analyse der Produktivumgebung verbunden.

SOA Service Management beim Provider

Zentrale Aufgaben des SOA-Service-Managements auf der Seite des Service-Providers sind Bereitstellung und Wartung bzw. Weiterentwicklung von Services, das Steuern der Nutzung von Service-Instanzen zur Laufzeit, die Erhebung von Key Performance Indicators (KPIs) und die Abrechnung der Service-Nurtzung.

Die IT Infrastructure Library (ITIL) hat sich als Quasi-Standard für das IT-Service-Management etabliert und liefert auch für das SOA-Service-Management das Rahmenwerk, um die oben genannten Anforderungen zu regeln.

Im Rahmen der Service-Delivery-Prozesse kann ein Provider beispielsweise einen fachlichen Service über unterschiedliche SLAs anbieten. Im Rahmen des Capacity-Managements werden je nach Bedarf dynamisch Ressourcen für die Service-Infrastruktur bereitgestellt (Virtualisierungslösungen können hier helfen, die Hardwarekosten nicht ins Unermessliche steigen zu lassen. Vor dem Einsatz von Virtualisierungslösungen ist jedoch zu klären, inwieweit die Lizenzbedingungen der Softwareplattform, auf deren Basis die Dienste realisiert werden, eine dynamische Ressourcenzuordnung zulassen oder ob die Lizenzierung auf der Basis der maximal zu nutzenden Ressourcen erfolgen muss.).

Beim Service Support müssen insbesondere die komplexen Service-Abhängigkeiten zwischen intern und extern erbrachten Services berücksichtigt werden. Die dynamische Service-Nutzung erschwert es zudem, die Auswirkungen von Störungen auf die Geschäftsprozesse zu ermitteln. Eine Verwaltung des Service-Netzwerkes wird insbesondere für die Business-Impact-Analyse, der Identifizierung von betroffenen Komponenten und Geschäftsprozessen, u. a. bei Störungen der zugrunde liegenden Infrastruktur oder beim geplanten Austausch von Services, benötigt. Für Störungen aus den Systemen heraus muss zudem das System-Management das Monitoring der eingesetzten SOA-Technologien berücksichtigen.

Zum Service-Management eines Service gehört zudem die Verwaltung der benötigten IT-Infrastruktur aus Hardware und Infrastruktur-Komponenten wie Datenbanken, Middleware etc. Der durch eine SOA gewonnene hohe Grad an Modularität und Flexibilität in der Applikationslandschaft verlangt nach entsprechenden Konzepten, die diese Modularität und Flexibilität auch auf der IT-Infrastrukturebene optimal ausnutzen. Lösungen und Technologien zur Virtualisierung der Infrastruktur unterstützen diese Anforderungen. Dabei kommt nur die Infrastruktur zum Einsatz, die zur Ausführung eines Service benötigt wird. Zudem werden bei erhöhter Last eines Services oder des Ausfalls einer Infrastruktur-Komponente automatisch die neuen benötigten Infrastruktur-Ressourcen beigefügt.

Zur Verwaltung der Services und der dazugehörigen Infrastrukturen werden dabei SOA Repositories oder eine erweiterte Configuration Management Database (CMDB) eingesetzt. Diese enthalten neben der Verwaltung von Services und der dazugehörigen Metainformationen auch die Verwendungsnachweise von Services aus anderen orchestrierten Services und von Geschäftsprozessen.

Veränderung der IT-Organisation

Die Zergliederung von monolithischen IT-Systemen und Anwendungen in einzelne Services und deren Zuordnung zu Geschäftsprozessen ermöglicht eine kontrollierte Auslagerung einzelner Services und Subprozesse. Darüber hinaus wird es so möglich sein zu kontrollieren, welchen Wert die Services für die jeweiligen Geschäftsprozesse des Unternehmens haben. Die sich daraus ergebenden Veränderungen für die IT-Organisation liegen in der höheren Messbarkeit der IT Leistungen und dem nachvollziehbaren Wertbeitrag der IT zu den Geschäftsprozessen des

Unternehmens. Der Aufbau eines Shared Service Center innerhalb der IT-Organisation ermöglicht die Umsetzung der oben genannten Service-Management-Aufgaben in Abstimmung mit den SOA-Governance-Prozessen. Das Service Center stellt die Wiederverwendbarkeit von Services im Unternehmen sicher. Darüber hinaus führt das Service Center die Performance-Kontrolle der verwendeten Services durch, erfasst und liefert die KPIs für die Optimierung der Geschäftsprozesse. Die Steuerung und Verrechnung der extern erbrachten Leistungen gehört bei extern bezogenen Services ebenfalls zu den Aufgaben. Das kommerzielle und technische Management von unterschiedlichen Service-Quellen sowie die Bereitstellung der KPIs für die Geschäftsprozesse ist die Hauptaufgabe der IT-Organisation für eine SOA.

5.2 Governance-Umsetzung

Governance für Service-orientierte Architekturen – Ein Überblick

Im Abschnitt 3.6 werden die Strategien und das Design der SOA Governance erläutert. Im folgenden Abschnitt wird darauf aufbauend erläutert, welche Konzepte für die Umsetzung einer SOA Governance relevant sind.

Das Zusammenspiel von Menschen, Prozessen und Informationen – verknüpft mit Applikationen auf verschiedensten Plattformen – erfordert die Koordination dieser „Ressourcen“ um den höchstmöglichen Nutzen zu erhalten.

Zu dieser Koordination gehört der Aufbau von Verantwortlichkeits- und Kommunikationsketten. So wird klar herausgestellt, wer für die jeweiligen Prozesse entscheidungsberechtigt ist. Hierbei ist zu beachten, dass Prozesse über Unternehmensgrenzen hinausgehen können.

Allgemein bezeichnet man die übergeordneten Strukturen zur Steuerung eines Unternehmens als Governance. Governance soll sicherstellen, dass alle Akteure und ihre Aktivitäten zum Erreichen der Unternehmensziele beitragen. Die Governance eines Unternehmens definiert: WAS getan wird, WIE es getan wird, WER es tut und WIE die Ergebnisse GEMESSEN werden. Entsprechend definiert Governance die Etablierung von Prozessen zur Definition von Entscheidungsrechten – wer ist berechtigt welche Entscheidungen zu treffen. Darüber hinaus werden im Rahmen der Governance Mechanismen der Erfolgsmessung und Kontrolle von Entscheidungen und deren Umsetzung etabliert.

Corporate Governance stellt sicher, dass jeder im Unternehmen gemäß der Unternehmensstrategie und den Geschäftsgrundsätzen handelt. Dies geschieht zum Beispiel über die Definition von Geschäftsprozessen mit zugehörigen Rollen und Verantwortlichkeiten.

IT-Governance betrifft entsprechend die IT eines Unternehmens. Sie umfasst die Prozesse sowie Rollen und Verantwortlichkeiten, die erforderlich sind, damit die IT die Geschäftsziele unterstützt.

SOA Governance erweitert die Prozesse der Corporate Governance und der IT-Governance. SOA Governance etabliert Governance-Strukturen und -Mechanismen, die im Lebenszyklus eines Service entscheidend sind – die Identifizierung von Services gehört dazu. Des Weiteren stellen sich in diesem Rahmen Fragen wie: Wer ist verantwortlich für einen Service, wer finanziert ihn oder wie wird ein Service entwickelt und implementiert. SOA Governance umfasst also sowohl Aspekte aus dem Bereich Corporate Governance als auch aus der IT-Governance.

Im Rahmen der Umsetzung Service-orientierter Architekturen werden bestehende Governance-Prozesse durch Konzepte für

- die Wiederverwendbarkeit von Einzelteilen
- das Verwalten der Lebenszyklen von Applikationsfragmenten und
- die Regelung von Abhängigkeiten innerhalb des Gesamtsystems erweitert.

Diese neuen Ansätze dienen als Erweiterung der IT-Governance und ersetzen diese nicht.

Die Ergebnisse der Koordinationsmaßnahmen werden in Service-Level-Agreements innerhalb der Prozesskette umgesetzt. Diese regeln die Anforderungen des Nutzers sowie die Leistungserbringung des Providers.

Governance beschäftigt sich mit Verantwortlichkeiten, Autorisierung etc., also mit dem „Wer?". Das Management hingegen definiert und überwacht die Prozesse, die hinter dem „Wer" stehen. Eine Studie des Analysten Gartner besagt, dass Governance für SOA-Projekte nicht optional, sondern zwingend notwendig ist. Ohne Governance fällt der Return on Investment gering aus. Darüber hinaus gefährdet die fehlende Governance Projekte bereits von der Pilot-Phase an [MAL 2006], da sich ohne entsprechende Steuerung häufig Kompetenzen überschneiden und keine effiziente Auslastung der Ressourcen erreicht wird.

Evaluierung von Geschäftsprozessen und Wiederverwendbarkeit

Wie bereits erwähnt ermöglicht eine SOA die Wiederverwendung von Teilprozessen (Services) und Anwendungsfragmenten. In diesem Kontext müssen Unternehmen ihre Services analysieren und priorisieren, so entsteht eine Übersicht, welche Services einen gehobenen Nutzen für das Unternehmen aufweisen und inwiefern diese als Standard (Basis) angesehen werden können. Innerhalb der SOA kann man einzelne Dienstleistungen bündeln und beliebig neu zusammenstellen, sodass beispielsweise auch gesetzliche Rahmenbedingungen in anderen Ländern eingehalten werden können, ohne komplette Prozessketten neu zu designen. Ihm Rahmen der Analyse der Teilprozesse wird auch untersucht, welche Applikationen durch welchen Prozess ausgelöst werden, sodass ein Zusammenspiel aller Unternehmenszweige und der entsprechenden Applikationen reibungslos verlaufen kann. Eine Voraussetzung für die Ausschöpfung der Vorteile (Wiederverwendbarkeit und Teilung von Ressourcen) einer SOA ist die Festlegung von Datenformaten (zum einheitlichen Informationsaustausch) und der Aufbau einer Bibliothek zur Verwaltung von wiederverwendbaren Prozessteilen und Applikationsfragmenten.

Umsetzung der SOA Governance

Wie für die Umsetzung einer SOA verspricht ein phasenorientiertes, stufenweises Vorgehen auch für die Umsetzung der SOA Governance die besten Erfolgschancen. Empfehlenswert ist beispielsweise ein Vorgehensmodell, das aus den folgenden vier Phasen besteht: Planung, Definition, Einführung, Erfolgsmessung. Diese vier Phasen werden iterativ immer wieder durchlaufen, sodass das Governance-Modell ständig erweitert und optimiert wird. Jede Phase besteht dabei aus genau definierten Zielen und Aktivitäten. Abbildung 31 stellt diese grafisch dar. Dabei wird stets auf bestehenden Strukturen und Mechanismen aufgebaut.

Eine SOA Governance sollte nicht nur mithilfe von Software etabliert werden. Eine effektive SOA Governance geht über den Einsatz von Technologie hinaus – sie fußt auf akzeptierten Prozessen und Rollen.

Neben diesem phasenorientierten Vorgehen hat sich ein weiteres Instrument als hilfreich erwiesen, SOA Governance erfolgreich zu etablieren: ein „SOA Center of Excellence (CoE)". Ein SOA CoE ist ein bereichsübergreifendes Gremium, das aus Mitgliedern der Fachbereiche und der IT-Bereiche auf unterschiedlichen Hierarchiestufen zusammengesetzt ist. Die Mitglieder des SOA CoE können Vollzeit oder Teilzeit für das Gremium arbeiten, permanent oder auf Rotationsbasis. Sie können als die „SOA-Keimzelle" des Unternehmens betrachtet werden. Zu den Aufgaben des CoE gehören neben der Unterstützung von Projekten durch SOA-Experten beispielsweise das Durchführen von Architektur-Reviews oder das Sammeln und Verwalten von Services und deren Metadaten. Abbildung 32 veranschaulicht mögliche Aufgaben eines CoE.

Durch seine Zusammensetzung aus Mitarbeitern der Fachbereiche und der IT sowie durch das Prinzip der Rotation unterstützt das CoE nicht nur den schnellen Wissensaufbau, sondern fördert auch die Akzeptanz von SOA im Unternehmen.

Planung

- Überprüfung der SOA-Vision, -Ziele und -Strategie
- Beurteilung bestehender Governancestrukturen und -mechanismen
- Bewertung vorhandener SOA-Fähigkeiten
- Untersuchung der Änderungsbereitschaft
- Definition von SOA-Governance-Zielen

Definition

- Definition von Standards und Policies
- Definition/Modifikation von Governanceprozessen, IT-Entwicklungs- und Betriebsprozessen
- Definition eines Modells für Service-Ownership und Service-Finanzierung
- Definition von Rollen und Gremien
- Identifikation unterstützender IT-Werkzeuge

Governance and Best Practices

Erfolgsmessung

- Überprüfung der Einhaltung definierter Vorgaben
- Überprüfung des Erfolgs von Organisational-Change-Maßnahmen
- Überprüfung des Governance-Modells und ggf. Einleitung von Anpassungen

Einführung

- Einführung der Rollen und ggf. des SOA Center of Excellence
- Einführung der Governance und Serviceleben-zyklusprozesse
- Kommunikation und Schulung von Strategie, Zielen, Standards, Policies etc.
- Implementierung der IT-Werkzeuge

Abbildung 31: Phasenorientiertes Vorgehen zur Umsetzung von SOA Governance

SOA-Bekanntheit und -Akzeptanz

Kommunikation von Zielen, Nutzen, Best Practices, Assets, Modellen, Standards, Methoden, Templates etc.

SOA-Vitalität und Thought Leadership

Kontinuierliche Analyse interner und externer Entwicklung und Anpassung der Architektur

Architekturreviews

Durchführung von unabhängigen Design- und Architekturreviews für Services

Betriebsunterstützung

Unterstützung des IT-Betriebs in Fragen der Implementierung, Performanceoptimierung, Monitoring etc.

SOA Center of Excellence

Projektunterstützung

Unterstützung von SOA-Projekten in Form von Fachexperten und Erfahrungen aus anderen Projekten

Wissenstransfer und -aufbau

Identifikation von Wissensbedarf und Entwicklung von Schulungsmaßnahmen

Asset Management

Verwaltung von Assets, Pattern etc., zur Reduzierung von Risiken und Beschleunigung von Projekten

Aufbau von Wissen und Assets

Bearbeitung von SOA-relevanten Fragestellungen, z. B. im Umfeld von SOA-Security

Abbildung 32: Mögliche Aktivitäten eines SOA Center of Excellence

Fazit

Worin liegt der Vorteil einer konsequenten SOA Governance?

Zunächst einmal lässt sich durch die Erfolgsmessung (Measurement-Phase) detailliert der Vorteil einer SOA-Infrastruktur mit den relevanten Daten benennen. Unternehmen, die interne Erfolgsmessungen durchführen, generieren einen 10–12 % höheren Market Value [McK 2002], da Investoren eher in Unternehmen investieren, die klare Prinzipien, Standards und Prozesse etabliert haben und diese darüber hinaus mit Zahlen belegen können. Auf dieser Basis ist es solchen Unternehmen möglich, Kosten zu senken und den Profit zu erhöhen. Insbesondere die Total Cost of Ownership (TCO) hinsichtlich der bestehenden IT-Infrastruktur fällt durch Wiederverwendung von Services, Prozessteilen und Applikationen positiver aus.

Ein Kernelement von SOA Governance ist die Einhaltung von SLAs im Unternehmen und über seine Grenzen hinaus: Prozessketten dehnen sich über Grenzen der Firmen hinweg und generieren Komplexität. Die konsequente Umsetzung einer Governance sorgt für Überschaubarkeit dieser Komplexität und die Sicherstellung der Qualität von Services.

Durch die Verknüpfung von IT- und Geschäftsprinzipien profitieren beide Bereiche von der Weiterentwicklung der anderen.

Da die Organisation der SOA Governance hierarchisch in das Unternehmen eingegliedert ist, entsteht kein künstlicher Bruch zwischen den Ebenen der Firmenkultur. Verantwortungen werden klar und übersichtlich zugeteilt, wobei das CoE eine zentrale Rolle bei der Bündelung von Kompetenzen einnimmt.
Zusammenfassend sollen hier die wichtigsten Fragen beantwortet werden:

Was genau ist SOA Governance?
– Die Etablierung von Entscheidungsrechten und Kontrollmechanismen für die Aktivitäten während des Lebenszyklus eines Service

Warum braucht man SOA Governance?
– SOA Governance ist entscheidend für Auschöpfung des Potenzials einer SOA.

- Wie wird SOA Governance eingeführt?
  – Für die Einführung von SOA Governance empfiehlt sich ein stufenweises Vorgehen basierend auf den vier Phasen: Planung, Definition, Einführung, Erfolgsmessung. Der Aufbau eines SOA CoE ist empfehlenswert.
- Wie passt SOA Governance zu bestehenden Modellen der Corporate-Governance und der IT-Governance?
  – Sind in einem Unternehmen bereits Governance-Strukturen und -mechanismen etabliert, so baut SOA Governance auf diesen auf und erweitert sie um SOA-spezifische Aspekte.
- Was sind entscheidende Erfolgsfaktoren für eine erfolgreiche SOA Governance?
  – Voraussetzungen für den Erfolg oder Misserfolg einer SOA Governance sind beispielsweise ein passendes Finanzierungsmodell, umfassende Unterstützung aus Fachbereichen und IT sowie dedizierte Ressourcen für die Durchführung der Governance-Aktivitäten.

Literatur

[MAL 2006] Service-Oriented Architecture Craves Governance, Gartner, Inc., Paolo Malinverno, January 20, 2006

[McK 2002] „A Premium for Good Governance“,
The McKinsey Quarterly, 2002, Nummer 3

5.3 Erweiterte Nutzenmodelle

Software as a Service

Was ist SaaS?

Mehr und mehr Menschen verfügen heute über einen breitbandigen Zugang zum Internet. Dabei entstehen immer neue Geschäftsmodelle im B2B sowie im B2C

Umfeld, die teilweise deutlich attraktiver sind als die bisher üblichen Modelle.

Software as a Service (SaaS) ist ein ideales Beispiel für so ein neues Geschäftsmodell. Anstatt kostspielige Softwareanwendungen zu kaufen und im eigenen Haus zu betreiben, gehen Unternehmen heute dazu über, das Nutzungsrecht an diesen Produkten, ähnlich wie bei Elektrizität oder Firmenwagen, bedarfsgerecht und befristet ohne jegliche Kapitalkosten und Aufwand zu erwerben.

Eine Studie von IDC [IDC 2005] fand heraus, dass "es häufig die Mitarbeiter der Fachbereiche sind, die mit Hilfe des SaaS-Modells ein Problem lösen, welches die IT-Abteilung zu diesem Zeitpunkt nicht lösen kann". Ähnlich wie der PC ermöglicht es jetzt das SaaS-Modell, die Vorteile der IT zu nutzen, ohne dabei gleichzeitig von der IT-Abteilung abhängig zu werden. Selbst recht komplexe Anwendungen lassen sich auf "Mietbasis" nutzen. Mit Hilfe des SaaS-Modells haben Unternehmen die Möglichkeit, bestimmte IT-Anforderungen sehr gezielt durch Outsourcing abzudecken. Während auf diese Weise bestimmte Anwendungen von SaaS-Anbietern bereitgestellt werden, bleiben die für den Unternehmenserfolg wichtigen Kernanwendungen im Haus.

Bei Gartner wird SaaS definiert als "Software, die sich im Besitz von einem oder mehreren Anbietern befindet und Dritten bereitgestellt wird". [GAR 2006] Das auch als "On-Demand Computing" oder "Application Service Provider" (ASP) bekannte SaaS-Modell existiert bereits seit mehreren Jahren. Im Gegensatz zu früher ist die Software heute jedoch über das Internet sehr viel besser zugänglich. Gemäß der erwähnten IDC-Studie erreichte der SaaS-Markt im Jahr 2004 weltweit ein Volumen von 4,2 Milliarden US$. Bei einer durchschnittlichen Wachstumsrate (CAGR) von 21 % wird das Volumen im Jahr 2009 auf 10,7 Milliarden US$ steigen [IDC 2005].

Laut Forrester [FOR 2005] zählen SaaS-Anwendungen in den Bereichen Vertriebs- und Marketingautomatisierung, Beschaffung, Personalwesen und Finanzen zu den häufigsten Anwendungen. IDC fand weiterhin heraus, dass der Trend zur Senkung der IT-Kosten sowie der Bedarf an umfangreichen Upgrades zu den wesentlichen Motivatoren für die Nutzung von SaaS zählen.

Demzufolge sind die Mehrzahl aller Unternehmen potentielle SaaS-Kunden. Gelegenheiten zur Vermarktung von SaaS-Lösungen gibt es ständig. SaaS eröffnet sogar noch zusätzliche Vertriebskanäle: Anstatt sich mit der IT-Abteilung auseinanderzusetzen, gelangen die SaaS-Anbieter in den direkten Kontakt mit den jeweiligen Fachabteilungen. Anstelle von langwierigen Diskussionen und Kostenanalysen lässt sich eine ganz einfache Frage formulieren: "Wird mit Hilfe der SaaS-Anwendung mehr Gewinn erzielt, als es die Nutzung des SaaS-Dienstes kostet?"

Welche unternehmerischen Vorteile ergeben sich aus der Nutzung von SaaS ?

Vor dem Hintergrund des zunehmenden Wettbewerbsdrucks sind viele Unternehmen dazu gezwungen, ihr Geschäftsergebnis durch Gewinnmaximierung und Streichung unnötiger Kosten zu optimieren. Dies ist leichter gesagt als getan, doch grundsätzlich gibt es immer ein offensichtliches Einsparpotenzial – nämlich alle Aktivitäten, die über den Bereich der Kernkompetenz des Unternehmens hinausgehen. Häufig werden Aufgabenfelder intern bearbeitet, die ohne Weiteres von externen Fachkräften erledigt werden könnten. Daran hat gewöhnlich auch die IT-Abteilung einen beträchtlichen Anteil.

Durch die Reduzierung der Kosten, die bereits vor der ersten Nutzung einer Anwendung anfallen, wird die Menge der potenziellen Kunden für SaaS drastisch erhöht. In diesem Zusammenhang stelle man sich einmal vor, wie viele Menschen pro Jahr einen Flug buchen und wie wenige sich dagegen den Luxus leisten, ein eigenes Flugzeug zu besitzen.

Die Anschaffung einer neuen Software ist für jedes Unternehmen ein extrem komplexer, teurer und zeitaufwändiger Vorgang. Dagegen lässt sich die gleiche Anwendung schnell, einfach und kostengünstig über SaaS nutzen. Bei vielen SaaS-Anbietern hat der Nutzer heute sogar schon die Möglichkeit, über einen einfachen Webbrow-

ser die gesamte Konfiguration „seiner“ Software selbst durchzuführen.

Die Vorteile für das eigene Unternehmen erkennen SaaS-Kunden meist schnell. Gleichzeitig wirkt sich jeder dieser Vorteile auch für den SaaS-Anbieter positiv aus, denn jeder zufriedene Kunde ist mit großer Wahrscheinlichkeit auch ein loyaler Kunde. Zu diesen beidseitigen Vorteilen zählen:

- Eine enge und dauerhafte Beziehung zwischen SaaS-Anbieter und Kunde
- Ständiges Feedback vom Kunden bezüglich zukünftiger Anforderungen sowie Problemstellungen
- Anstelle umfangreicher neuer Softwarereleases in langen Abständen von 12 oder 18 Monaten können SaaS-Anwendungen deutlich häufiger aktualisiert werden

SaaS ist somit ein Meilenstein für ein neues Marktsegment. Wie schon in der Vergangenheit in anderen Bereichen werden auch hier die Anbieter die größten Marktanteile erobern, die diese Nische zuerst erkannt und besetzt haben. Sie können in der ersten Phase so lange günstige Preise anbieten, bis ein richtiger Wettbewerb eingesetzt hat. Auch nach dieser Phase sind sie (vorausgesetzt sie haben sich gut positioniert) meist noch diejenigen, nach denen sich die Wettbewerber richten müssen.

Praktische Umsetzung von SaaS

Gartner weist darauf hin [GAR 2006], dass alle Unternehmen, die den Einsatz von SaaS erwägen, potenziellen Anbietern eine Reihe sehr gezielter Fragen stellen sollten. Einige dieser Fragen haben mit der Anwendung selbst zu tun, andere dagegen mit der eingesetzten Infrastruktur. Hier einige Beispiele:

Konfigurierbarkeit: SaaS-Anwendungen sind keine Fertig-Lösungen von der Stange, die alle Anforderungen abdecken. Häufig sind anwendungsspezifische Anpassungen erforderlich. Daher ist es immens wichtig, dass der Anbieter auf Änderungs- und Erweiterungswünsche des Kunden wie z. B. geänderte Datenmodelle oder neue Benutzeroberflächen schnell und effektiv reagieren kann.

Integration: Unternehmensanwendungen sind wenig effektiv, wenn sie als isolierte Lösung oder „Insel der Automatisierung“ betrieben werden. Die Mehrzahl der Unternehmen würde ihre CRM-, ERP- oder andere Lösungen gern an bestehende Anwendungen oder Datenbanken anbinden. Im Idealfall kann eine solche Integration schnell und flexibel bei minimaler Auswirkung auf den Alltagsbetrieb umgesetzt werden.

Berichterstellung und Analyse: Neben den üblichen Abfragen sind Berichterstellung und Analyse ein besonders wichtiger Bestandteil der Benutzeranforderungen. SaaS-Infrastrukturen, die diese Anforderungen erfüllen, erhalten eine große Akzeptanz – besonders dann, wenn sie gut integriert und leicht anzupassen ist.

Anwenderzufriedenheit: Gartner weist darauf hin, dass “die Anwenderzufriedenheit ein wichtiger Maßstab für den Erfolg einer Anwendung ist ”. Zwar ist es oft schwer, die Anwenderzufriedenheit im Detail genau zu erfassen. Zweifellos zählen hier jedoch Zuverlässigkeit, schnelle Reaktionszeit und Konsistenz zu den wichtigsten Kriterien.

Welche Art von Infrastruktur benötigt nun ein typischer SaaS-Anbieter? Zunächst muss diese Infrastruktur extrem leistungsfähig sein. Egal wie die Kundenanforderung lautet – ob neue Unternehmensanwendung, die Erweiterung bestehender Applikationen um ein Web-Interface, die Integration mit bestehenden Datenbanken und Programmen oder die Umsetzung zusätzlicher Funktionalität – entscheidend ist eine schnelle, reibungslose und zuverlässige Umsetzung.

Geschäftsmodelle und Datenstrukturen sollten auf einer möglichst hohen Abstraktionsebene definiert werden. Die daraus hervorgegangene Software sollte auf möglichst viele Plattformen portierbar sein. Alle wichtigen Standards und führenden Softwareprodukte sollten unterstützt werden – zum Beispiel Betriebssysteme, DBMS, Programmiersprachen, Middleware, Webserver

und Browser. Beim Geschäft mit SaaS sind jegliche Einschränkungen der kundenseitig einsetzbaren Hardware oder Software nicht akzeptabel. Auswahlmöglichkeiten sollten so transparent wie möglich dargestellt werden. Als Minimalanforderung sollte jeder Kundenrechner mit den Betriebssystemen Windows, Unix oder Linux, oder Macintosh lauffähig sein, optional mit Java und einfachen Webbrowsern.

Für den Anwender muss es leicht sein, seine persönliche Nutzeroberfläche und den Zugang zum Service an seine Bedürfnisse anzupassen – zum Beispiel das gewünschte „Look-and-Feel", die Anordnung wichtiger Dokumente und Datenquellen oder die Erzeugung individueller Abfragen und Berichte. Diese Art von Flexibilität erspart Entwicklern und Administratoren viele zeitaufwändige Anfragen, die produktivitätsmindernd wirken. Die Anwendungen müssen einfach bedienbar sein sowie konsistente Schnittstellen anbieten. Weiterhin sollten sie störungsfrei mit guten Antwortzeiten laufen und unter der Oberfläche gut integriert sein.

SaaS-Anwendungen sollten weiterhin über einen durchgängig hohen Grad an Sicherheit verfügen, damit jeder Nutzer nur auf genau die Daten zugreifen kann, für die er eine Berechtigung erhalten hat. Die Systemverwaltung sollte so einfach wie möglich gehalten werden und erfolgt im Idealfall zu 100 % über das Internet.

Um Hindernisse während der Eingewöhnungsphase so niedrig wie möglich zu halten und den Service für eine möglichst große Anzahl potenzieller Kunden zu erschließen, sollte ein Einstieg mit einem einzelnen, kostengünstigen Server bei späterer bedarfsorientierter Kapazitätserweiterung möglich sein. Bei starker Nutzung sollte die ideale SaaS-Infrastruktur durch Spezialisierung und mengenabhängige Skalierbarkeit der Serverinfrastruktur eine schnelle Verarbeitung ermöglichen.

Schlussfolgerung – Ausblick auf SaaS

Es kann kein Zweifel darüber bestehen, dass sich SaaS für Softwareanbieter zu einem zunehmend wichtigen Geschäftsfeld entwickelt, besonders in den Bereichen Automatisierung von Marketing und Vertrieb, CRM, Beschaffung, Personalwesen und Finanzen. Da Anschaffung, Installation und Betrieb teurer Hardware- und Softwareprodukte nicht mehr erforderlich sind, haben Leiter von Fachbereichen die Freiheit, sich eigenständig für bestimmte Lösungen zu entscheiden. Dabei ist lediglich sicherzustellen, dass Nutzen und Kosten einer Anwendung in einem optimalen Verhältnis stehen und der betreffende Anbieter den benötigten Service-Level vertraglich abgesichert anbieten kann.

Wie bereits erwähnt eröffnet SaaS einen großen Markt für Softwareanbieter. Die Vermarktung von SaaS-Anwendungen unterscheidet sich dabei wesentlich vom Vertrieb von Softwarepaketen für zentrale IT-Abteilungen. Die Hürden für einen Markteinstieg sind hier deutlich niedriger angesetzt, da Anwendungen schneller bereitgestellt werden können.

Literatur

[IDC 2005] Market Analysis: Worldwide and U.S. Software as a Service 2005-2009 Forecast and Analysis", IDC, March 2005

[GAR 2006] "Key Issues for Software as a Service, 2006", Gartner ID Number: G00138654, March 2006

[FOR 2005] "Software-As-A-Service: Empowerment for the Business User, Potenzial Havoc for IT", Forrester Research, Inc., June 2005

[GAR 2006] Evaluating Software-as-a-Service Providers: Questions to Ask Potential SaaS Providers", Gartner ID Number: G00138777, April 2006

6 Ausblicke

6.1 SOA – kurzfristiger Trend oder tief greifende langfristige Veränderung?

Mark Twain wird der Satz zugeschrieben, dass Vorhersagen schwierig seien, besonders wenn sie die Zukunft betreffen. Die Diversität der heute publizierten Ausblicke zum Thema SOA scheinen diese Ansicht zu bestätigen: Nichts ist derzeit schwieriger, als einen Ausblick auf die zukünftige Marktentwicklung von Service-orientierten Architekturen zu geben.

Es erschien dem maßgeblich an der Erstellung des Leitfadens beteiligten BITKOM Arbeitskreis daher wichtig, dem Leitfaden einen zwischen allen Teilnehmern abgestimmten, einheitlichen Ausblick hinzuzufügen.

SOA besteht nicht nur aus Technologien. SOA ist vielmehr ein „Mindset", eine Kultur, eine Transformation. Vor allem aber ist SOA ein längerfristiger Ansatz, Geschäftsabläufe zu optimieren, zu flexibilisieren und Geschäft (Business) und IT näher zusammenzubringen. Service-orientierte Architekturen werden als notwendiger Schritt angesehen, die IT als Wegbereiter und Beschleuniger für das zukünftige Geschäft zu positionieren.

SOA-Projekte werden mit der Zeit durch Zunahme und häufige Wiederverwendung von Services immer besser und erhöhen die Flexibilität für viele Bereiche. Je mehr der Endanwender die Software nutzt, desto besser kann sie werden. Deswegen sind Service-orientierte Ansätze und Projekte – laut der Meinung führender Analysten und deren Kundenumfragen – eine „ongoing journey", sie verfolgen ein langfristiges Ziel: im Business- und IT-Bereich so flexibel und im Anwendungsbereich so wiederverwendbar wie nur möglich zu werden. Flexibilität in IT und Business sowie Wiederverwendbarkeit von Services werden nur durch standardisierte Formate und Schnittstellen erreicht.

Unserer Ansicht nach

- wird sich der SOA-Ansatz in vielen Aspekten über einen Zeitraum von 5 Jahren entscheidend weiterentwickeln und reifen.
- werden die in diesem Leitfaden beschriebenen SOA-Kernthemen von allen Teilnehmern als gemeinsamer Architektur-Standard weiterhin unterstützt und fortentwickelt werden.
- werden neue Formen der Zusammenarbeit (z. B. Kollaboration im Web 2.0) und verfeinerte Methoden den SOA-Standard durch auf dieser Basis erweiterte Nutzenpotentiale stabilisieren.
- wird SOA die Grundlage für eine neuartige Zusammenarbeit zwischen IT und Fachabteilung, zwischen Kunden und Softwareanbieter sowie zwischen Kunden und der IT-Serviceindustrie.
- wird SOA eine Industrialisierung der IT vorantreiben.

SOA – quo vadis?

Doch wo geht die Service-orientierte Reise hin? In proprietäre IT-Umgebungen? In proprietär modellierte Geschäfts-Services? In proprietäre Software-Architekturen und Technologien? Genau der Gegenteil sollte der Fall sein. In den Medien wird oft über ein Scheitern des SOA Ansatzes berichtet. Alle Meldungen über das allgemeine Scheitern von begonnenen SOA-Projekten oder sogar des SOA-Ansatzes in der IT halten wir für unzutreffend. Der Kern des heutigen SOA-Ansatzes wird allen Versuchungen widerstehen, proprietäre Sonderwege einzuschlagen, und somit stabil bleiben.

Die SOA-Kernthemen umfassen unserer Meinung nach

- eine neue Art der Zusammenarbeit zwischen Fachabteilung und IT auf Basis eines einheitlichen SOA-Ansatzes
- eine auf Standards basierende Interaktion unterschiedlichster IT-Komponenten
- die Verwendung von offenen Web Services und offenen Schnittstellen, welche die Kommunikation zwischen den einzelnen Services und deren

Wiederverwendbarkeit durch herstellerübergreifende Web-Service-Standards gewährleisten werden.

Unsere Meinung basiert auf der in diesem Leitfaden ausführlich erläuterten Erfahrungen, dass SOA eine umfassende Antwort auf dringende Probleme der heutigen IT-Märkte bietet:

- Heutige IT-Landschaften sind extrem komplex und bieten keine neuen Lösungen.
- Die Aufwände für Wartung und Erhaltung bestehender Infrastrukturen lässt zu wenig Ressourcen für Erweiterungen und Innovation übrig.
- Der Unzufriedenheit von Fachabteilung und IT wächst, da mit den bestehenden IT-Landschaften das gewünschte Tempo der Veränderung im Unternehmen nicht realisiert werden kann, zum Nachteil aller Beteiligten.
- IT-Budgets werden in das Kostenmanagement mit einbezogen. Von IT-Bereichen werden deutliche Produktivitätsfortschritte eingefordert: Mit weniger Ressourcen mehr Produktivität zu erreichen.

SOA bietet für diese dringenden Probleme einen Lösungsansatz. Es ist daher zu erwarten, dass SOA für die nächsten 5–7 Jahre ein zentrales Thema für IT-Kunden und die gesamte IT-Industrie sein wird.

Weiterentwicklungen der SOA, die zurzeit unter Begriffen wie Semantic SOA, event-driven SOA, Enterprise SOA diskutiert werden, werden an den grundlegenden Paradigmen nichts ändern, sondern bestenfalls das aktuelle Konzept von SOA erweitern und ergänzen.

Befürchtungen der Kunden dass die heute auf gemeinsamen Standards basierende SOA in herstellerspezifische proprietäre SOA-Angebote zerfallen wird, lassen den Wunsch der Kunden nach einheitlichen Standards und die mittlerweile beachtliche Durchsetzungskraft der Kunden außer Acht. Die meisten Unternehmen haben eine heterogene IT-Landschaft von mehreren IT-Anbietern, proprietäre SOA-Ansätze lösen die beschriebenen Problem nicht und werden daher keine Käufer finden.

Als Mark Twain seine eigene Todesanzeige in einer Zeitung lesen musste, die ein Witzbold dort plaziert hatte, antwortet er wie folgt: *„Alle jüngsten Aussagen über mein Ableben halte ich persönlich für stark übertrieben.“*

Die Kräfte des Marktes werden SOA als gemeinsamen Standard stabilisieren und zur dominierenden Architektur werden lassen.

6.2 Auf Basis von SOA entstehen neue Formen der Zusammenarbeit

SOA-Services können von jedermann in jeder ausführbaren Computer-Sprache erstellt und angeboten werden. SOA setzt die Standards, die alle IT-Komponenten erfüllen müssen, damit sie auf allen SOA-Plattformen ablauffähig sind. Neue Anwendungen können entstehen, ohne dass fundierte Kenntnisse über Plattformen notwendig sind.

Alle namhaften Hersteller von IT-Plattformen setzen schon heute auf SOA – sowohl im Bereich Middleware als auch im Bereich der Anwendungen. Sie entwickeln die neusten Versionen ihrer Software auf Basis von SOA und fordern auch Fremdanbieter auf, zusätzliche Services anzubieten. So werden sich um diese „Superplattformen“ sehr lebendige SOA-Ökosysteme entwickeln, die neue, industriespezifische „composite Business-Services“ anbieten werden, sowohl für Anwendungen, die jeder Mitarbeiter brauchen kann, als auch für spezielle Industrielösungen. Diese Ökosysteme, auch „Business Services Fabric“ genannt, sind sicherlich ein wesentliches Differenzierungsmerkmal im Wettbewerb der Anbieter und werden den Kunden zusätzlichen Nutzen bringen.

Innovative kleine Start-ups haben in der SOA-Welt ebenfalls große Chancen, sich mit ihrem Mehrwert zu etablieren, und ihre Services auf einfache Weise in die IT-Welt ihrer Kunden zu integrieren.

Neue Impulse werden auch durch sogenannte Mashedups entstehen, d. h. durch die Verknüpfung von (IT- oder Kommunikations-)Anwendungen mit Diensten aus dem Web. So können neue Dienste mit Mehrwert entstehen,

die z. B. die präsenzbasierte Kooperation um geographische Daten erweitert, um Treffpunkte festzulegen und den Weg zu den Meetings effizient zu planen.

Unternehmen, die im Rahmen des SOA-Ansatzes über ihre Grenzen hinaus zusammenarbeiten, können Fremdservices nutzen und von der Infrastruktur ihrer Partner profitieren, sodass eine engere und bessere Kooperation als bisher ermöglicht wird.

Unternehmenskritische Services können mit Services gekoppelt werden, die von Application Service Providers (ASPs) oder Independent Software Vendors (ISVs) angebotenen werden. Dieses Vorgehen kann große Kostenvorteile erzeugen, ohne dass wesentliches Unternehmens-Know-how ausgelagert werden muss.

Die Wahlmöglichkeiten der Unternehmen erweitern sich sehr: Sie können auf bewährte Partner setzen und darüber hinaus neue Services integrieren, die ihr eigenes Angebot unverwechselbar machen. Insbesondere die Unterstützung neuer Geschäftsmodelle ist in diesem Kontext ein wichtiger Punkt.

Damit ergeben sich weitere mögliche Auswirkungen von SOA:

In IT-Programmen umgesetztes Fachwissen wird über die technischen Grenzen einer Plattform hinaus für alle SOA-Plattformen verfügbar.

- Das Angebot an Komponenten für SOA steigt
- Der potentielle Absatzmarkt für SOA-Komponenten wächst
- Der wirtschaftliche Anreiz SOA-Komponenten anzubieten, ist größer als bei einzelnen herstellerspezifischen Plattformen.
- SOA-Kunden haben mehr Auswahl an Services und Service-Providern

Anwendungssoftware wird fein-granularer.

- Auf Basis einer SOA-Plattform sind Komponenten unterschiedlicher Herkunft den SOA-Standards folgend einfach integrierbar. Eine integrierte Paketierung der Anwendungen wie heute üblich, wird an Bedeutung verlieren.

Geschäftsprozesse und Abläuf der Betriebwirtschaft werden zunehmend standardisiert, um Komponenten und Inhalte eindeutig beschreiben zu können

- Eine einheitliche Beschreibungen von SOA-Komponenten ermöglicht es, einzelne Komponenten separat anzubieten
- Neue Angebotsformen und Abrechnungsmodelle entstehen
- Eine globale Industrie für Komponenten entsteht
- Komponentenhersteller aller Größenordnungen können mit gleichen Chancen in einen globalen Wettbewerb treten
- WEB2-ähnliche Communities entstehen, sie treiben die Entwicklung für Komponenten und Anwendung der SOA voran

Grenzen zwischen Kunde, Softwareanbieter, Serviceanbieter, Communities etc. werden aufgehoben:

- Kunden, Servicefirmen, Communities etc. können ihr Fachwissen in einer Komponente bündeln
- Der Fokus der Kunden sowie der von SOA Anbietern wird sich von der SOA-Infrastruktur hin zu ganzen Service-Angeboten auf Basis von SOA verlagern
- Die Fachkompetenz entscheidet über den Erfolg einer Komponente
- Neue Formen der Zusammenarbeit im Markt entstehen
- Zahlreiche Aus- und Neu-Gründungen im Bereich der Komponentenanbieter werden entstehen

SOA ist somit – wie die gesamte IT-Branche – ständig in einem Wandel. Doch welchen Einfluss haben neue IT-Technologien auf zukünftige SOA-Aktivitäten und Projekte ?

6.3 Erweiterter Ausblick unter Einbeziehung anderer Technologien

Einfluss neuer IT-Technologien auf zukünftige SOA Projekte

Laut einer Umfrage unter den Top 500-Firmen werden deren zukünftige SOA-Aktivitäten (gemäß einer AMR-Studie in 2006) von folgenden Themen beeinflusst:

Verbesserte Integration von Informationen und Services (IaaS)

IaaS (Information as a Service) macht Informationen in Service-orientierten Anwendungen konsistenter und erreichbarer. Ziel von IaaS ist es, Informationen generell als Service zur Verfügung zu stellen, damit jeder Service-Nutzer in Echtzeit regelbasiert und vor allem kontrolliert auf aktuelle Informationen konsistent zugreifen kann. So wird eine hohe Datenqualität für Services sichergestellt. Der Trend geht hier zu Daten-Services aus verschiedenen SOA-basierten Anwendungen und Portalen, die konsistent sind und auf die in Echtzeit zugegriffen werden kann. Sie sind mit anderen Services kombinierbar, um neue Geschäftsprozesse und Daten zu erzeugen.

Größere Kundenorientierung durch Stammdaten-Management

Gerade durch den granularen Aufbau einer SOA hat sich der Wunsch der Fachabteilungen nach einer konsistenten, vollständigen und vor allem hochaktuellen Sicht auf wichtige Stammdaten (Kunden, Partner, Lieferanten, Produkte) noch mehr verstärkt. In einer SOA spielt gezieltes Stammdatenmanagement (Master Data Management) eine besondere Rolle: Über definierte Information-Services werden wichtige Geschäfts- und Stammdaten stets aktuell, konsistent und standardisiert anderen Geschäftprozessen bzw. Services zur Verfügung gestellt, z. B. durch andere Benutzer, Anwendungen oder Prozesse. Innovative Lösungen führender Hersteller für das Master Data Management basieren heute auf einer SOA und verbinden Geschäftsabläufe/Geschäftsprozesse mit zentralen Geschäftsdaten (Stammdaten) durch Informations-Integration (SOI-, SOA-Integration). Dieser Integrations-Prozess wird auch „Master Data Services" (MDS) genannt und sorgt für die Konsistenz, Vollständigkeit und Aktualität von wichtigen Stammdaten. Darüber hinaus ermöglicht er die Pflege von Daten aus heterogenen Quellen. MDS im Rahmen einer SOA ermöglicht eine umfassende Sicht auf Kunden und Produkte und verbessern den Service-Level für Kunden und Partner. Darüber hinaus wird die Einhaltung und das Management gesetzlicher Anforderungen (Compliance) erleichtert.

Standardisierung von Web Services

Heute werden Web Services hauptsächlich dazu verwendet, über interne, vor allem aber externe Integration von SOA-Komponenten einen Business Value zu erzeugen. Die Entwicklung geht dahin, „diskrete" Prozesse und Applikationen als Services von externen Providern zu beziehen, was wiederum den Trend zum Provisioning und zur Orchestrierung von Geschäftsprozessen beschleunigen wird.

End-to-End XML:

Ein Kern-Element einer SOA ist die Vereinbarung und Verwendung von standardisierten Schnittstellen (Interfaces) zwischen Service-Nutzern und Service-Anbietern. Bevor XML-fähige relationale Datenbanken existierten, war es mit höherem Programmieraufwand verbunden, relationale und XML-Datenstrukturen zu vereinen. Durch die neuen XML-Funktionalitäten und die Möglichkeiten, Daten in dem XML-Format zu speichern und zu übertragen, haben moderne relationale Datenbankmanagementsysteme (RDBMS) das zu übertragende Datenvolumen und somit die Aufwände für Speicherung, den Austausch und die Transformation der Service Messages reduziert. Für SOA-Umgebungen bedeutet dies eine echte End-to-End-Verwendung von XML-Nachrichten sowie den gezielten Austausch dieser XML-Nachrichten über Web Services.

Standardisierung der Business Process Execution Language (BPEL):

Die intelligente Automatisierung von skalierbaren und komplexen Geschäftsprozessen mittels der Kombination verschiedener Software-Technologien sowie die Möglichkeit der internen wie auch externen Kommunikation über standardisierte Services mit verschiedenen Geschäftspartnern macht SOA besonders wertvoll. Eine SOA wird dann noch einfacher zu realisieren sein, wenn mithilfe

ausgereifter und kompletter BPM-Lösungen, die auf einem gemeinsamen BPEL-Standard aufbauen, Geschäftsprozesse über Standard-Services herstellerübergreifend und produktunabhängig miteinander kommunizieren können.

Erweiterung der SOA durch Event-Driven Architecture (EDA):

Im Rahmen einer ereignisgesteuerten Software-Architektur wird die Serviceorientierung einer SOA mit einer ereignisorientierten Verarbeitung von Services und Daten in Echtzeit kombiniert.

Die Standards bezüglich Business-Modellierungs-Sprachen sind weitgehend gesetzt, z. B. durch BPEL bzw. BPEL4WS, doch für EDA hat sich ein Web-Services-Standard wie z. B. Web Services Eventing noch nicht durchgesetzt. Dies ist auch der Grund, warum EDA mit SOA in vielen Projekten bisher noch wenig Praxisanwendung gefunden hat. Die Erweiterung von SOA-Projekten durch EDA bringt eine zusätzliche Komplexität mit sich, für die weitreichende Governance-Mechanismen notwendig werden.

Aktive SOA Governance und Business Activity Monitoring durch den gezielten Einsatz einer Service Registry bzw. eines Repository (auch Service Repository Federation)

Viele SOA-Projekte sind bisher wegen unzureichender oder fehlender Governance-Aktivitäten nicht erfolgreich verlaufen. SOA Governance ist primär ein Prozess des Service Lifecycle Managements, der nicht nur alle Phasen eines SOA Deployments, sondern auch die der Ausführung von Prozessen überwacht – bis hin zu Business Activity Monitoring.

Bei einer optimalen SOA-Governance-Strategie ist die Verwendung einer Service Registry und Repository für SOA immens wichtig: Ein SOA Registry und Repository unterstützt den gesamten Lebenszyklus von Prozessen, Policies und Services. Während Service Registries meistens Schnittstellenbeschreibungen von Diensten speichern, sind Repositories für die Verwaltung einer Vielzahl weiterer Service-Informationen vorgesehen. Eine SOA Registry und Repository wird manchmal zu einer integrierten Einheit zusammengefasst und auch als „SOA Repistry bezeichnet.

Anwendungen werden in einer SOA durch Services ergänzt – so werden viele neue Services geschaffen, die freizugeben und zu steuern sind. Dadurch entstehen komplexe Abhängigkeiten und Netzwerke der Services untereinander, was wiederum ein gut geführtes (Business) Services Registry und Repository notwendig macht. Die Integration von verschiedenen Repositories und der föderierte Zugriff auf diese, um SOA Assets schneller identifizieren, verwalten und wiederverwenden zu können, wird die Leistungsfähigkeit einer SOA positiv beeinflussen, da mehr Services kombiniert und neue Services geschaffen und verwendet werden können. Business Services Repositories sind der organisatorische Mittelpunkt einer SOA-Lösung. Durch den Anstieg der gesetzlichen Bestimmungen und Compliance-Anforderungen werden SOA Governance und Business Activity Management immer bedeutender – ihnen obliegt es, für die Übersichtlichkeit und Transparenz von Geschäftsabläufen und Prozessen zu sorgen. So müssen im Rahmen von SOA zukünftig Geschäftsregeln (Business Rules) etabliert werden, um eine Business Integrity (Geschäftsintegrität) zu gewährleisten.

Darüber hinaus gilt es, ein Geschäfts-Vokabular (Business Vocabulary) zu entwickeln sowie eine erweiterungsfähige Business-Reporting-Sprache (Business Reporting Language), XBRL, zu etablieren.

Kopplung geschäftsbezogener und technischer Metadaten (Metadata Management)

Metadaten sind entscheidend für SOA Governance, für die Wiederverwendung von Services und für die Agilität eines Unternehmens durch SOA.

Die Services beschreibenden Metadaten werden in einer komplexen Service-orientierten Architektur immer wichtiger, um die Vielzahl der bereits existierenden Services klar voneinander abzugrenzen, den Geschäftskontext und die technologischen Assets miteinander zu verknüpfen und schließlich um IT und Geschäftsziele in Einklang zu bringen.

Einsatz industriespezifischer SOA-Frameworks und -Methoden

Industriespezifische, vorgefertigte SOA-Pakete beschleunigen für bestimmte Geschäftsbereiche die Entwicklung einer neuen Art von Service-orientierten Anwendungen, den sogenannten Component Business Services (CBS). Eine Orientierung der SOA an fachlichen Geschäftsfeldern (sog. Industry Solution Packs bzw. Frameworks) ist für ihren Erfolg wesentlich: Der unmittelbare Bezug zum Geschäftsfeld wird hergestellt, industriespezifische, vordefinierte und somit standardisierte Prozesse können verwendet werden und technologische und geschäftliche Anforderungen der jeweiligen Branche werden so durch die SOA-Lösung näher zusammengebracht. In diesen Frameworks sind Geschäftsprozesse in Form von industriespezifischen Services auf die jeweiligen Branchen zugeschnitten, können aber auch auf spezielle Anforderungen einer Branche angepasst werden.

Business Blueprints und SOA Industrie Frameworks werden miteinander gekoppelt, um Umsetzung der Geschäftssicht eines Unternehmens durch die IT effektiv zu steuern und die Transformation der Geschäftsbereiche und ihrer IT-Systeme zu einem Service-orientierten operationalen Modells zu unterstützen. Zu diesem Zweck werden Component Business Services verwendet. Diese erzeugen neue Funktionalitäten, indem Service-Komponenten aus verschiedener Quellen zusammengeführt werden.

Entwicklung neuer und Weiterentwicklung existierender Geschäftsprozesse durch die gezielte Bündelung bestehender Services zu neuen Anwendungen (Composite Applications)

Sogenannte „Composite Applications“ ermöglichen die Abbildung von komplexen Geschäftsabläufen, die durch spezielles Packaging oder als Appliances gebündelt, also vereinfacht, wurden. Die stetige Weiterentwicklung und Verbreitung von webbasierten Services führt zu einer effektiven Komposition von Business Services. Gezielt zusammengestellte Geschäftsabläufe lassen sich in einzelne Services zerlegen und wieder neu zusammensetzen, um sie hinsichtlich Effizienz und Flexibilität zu optimieren. Hier ist es wichtig, das Maß für Zerlegung und Wiederzusammensetzung von Services zu wählen und Standard-Schnittstellen für deren Kommunikation zu definieren.

Unternehmensportale als Einstiegspunkt und Interaktionsservices verschiedener SOA-Komponenten

Flexible Unternehmensportale bekommen gerade durch eine SOA eine immer größere Bedeutung, da sie als Einstiegspunkt und für Services für direkte Interaktion des Benutzers mit dem System verschiedene SOA-Komponenten und -Architekturen integrieren (Web 2.0-Komponenten, Composite Applications, Real-time-Informationen, konsistente Kunden- und Stammdaten). So erhöhen sie die Benutzbarkeit („Consumability“) einer SOA für den Anwender deutlich. Portale haben prinzipiell das Ziel, durch die Komposition bestehender Services, Endanwender ständig noch produktiver zu machen.

In Portalen entwickeln sich die Schnittstellen zum Endanwender durch den Einsatz von konfigurierbaren Portlets schneller und leisten so einen stetig wachsenden Wertbeitrag zur Geschäftsoptimierung (Beispiele: Wikis, Mashups, Blogs, RSS Feeds). Daraus ergibt sich eine höhere Flexibilität und Agilität für das Geschäft, da die Unternehmensintegration über neuartige Kollaborationstools gefördert wird.

Fazit

Mit der Entwicklung eines „Ökosystems“ auf Basis einer SOA enststeht ein erheblicher Mehrnutzen. SOA wird eine ähnlich dynamische Entwicklung wie das Internet nehmen. Auf Basis der Komponenten entstehen industrielle Muster (Patterns), die in anderen (älteren) Industrien wie Maschinenbau, Automobilbau, Chemie- und Pharma-Industrie bekannt sind und sich bestens bewährt haben.

7 SOA und Security

7.1 Einleitung und Abgrenzung

Allgemein steht der Begriff Informationssicherheit für Eigenschaften von informationsverarbeitenden und -lagernden Systemen, welche die Vertraulichkeit, Verfügbarkeit und Integrität von Daten und Prozessen sicherstellen. Mögliche Angriffe, Industriespionage oder unbefugte Zugriffe auf vertrauliche Daten sollen verhindert werden. Informationssicherheit dient also dem Schutz vor Gefahren, der Vermeidung von Schäden und der Minimierung von Risiken.

Informationssicherheit und SOA – welche neuen Herausforderungen gibt es?

Welche Aspekte im Bereich Informationssicherheit verlangen im Kontext Service-orientierter Architekturen besondere Aufmerksamkeit?

Generell erfordert SOA keine gesonderten oder neuen Techniken bzw. Konzepte zur Sicherstellung von Informationssicherheit. Es liegt allerdings auf der Hand, dass Systeme und Architekturen, die auf offenen, weltweit gut bekannten Standards aufsetzen, ohne entsprechenden Schutz erheblich anfälliger für unbefugte Eingriffe sind als Systeme der Vergangenheit, welche nur von wenigen Experten weltweit verstanden werden. Diese sogenannte „Security by Obscurity“ ist also kein effektiver Schutz mehr für offene vernetzte Architekturen wie Internet oder SOA.

Bei der Einführung einer SOA muss man sich deshalb hundertprozentig auf die etablierten Sicherheitstechniken verlassen können. Bei der Überführung einer Anwendungslandschaft in eine SOA sollte daher eine Revision der Maßnahmen für die Informationssicherheit immer fester Bestandteil des Projektes sein.

Service-orientierte Architekturen sollen eine schnelle Anpassung der IT an sich verändernde Geschäftsprozesse ermöglichen. Diese Agilität bedeutet, dass auch die Sicherheitskonzepte flexibel gestaltet werden müssen. Derartige Herausforderungen ergeben sich beispielsweise durch die Einbindung von Partnern und Kunden in die eigenen Systeme eines Unternehmens. Dabei muss die Sicherheit jedoch immer beherrschbar und bezahlbar bleiben.

Der vorliegende Leitfaden soll bei der Erarbeitung von geeigneten Lösungen und Security-Konzepten unterstützen. Die hier dargestellten Anregungen und Erfahrungen helfen bei der Analyse des Sicherheitsbedarfs und der Erarbeitung eines geeigneten Sicherheitsentwurfs. Darüber hinaus finden sich hier Anregungen, um gegebenenfalls in laufende SOA-Vorhaben korrigierend einzugreifen. Der Fokus dieses Leitfadens liegt auf SOA-spezifischen Security-Aspekten. Eine Beschäftigung mit allgemeinen Security-Aspekten ist in diesem Rahmen unerlässlich und muss selbstverständlich auch Bestandteil von SOA-Projekten sein.

7.1.2 Security-Aspekte: Grundsätzliche Security-Ziele

Auch im Rahmen von Geschäftsprozessen bleiben die klassischen Security-Ziele aktuell. So muss zum Beispiel sichergestellt werden, dass Hard- und Software erwartungsgemäß funktionieren. Darüber hinaus müssen Vorsorgemaßnahmen getroffen werden, damit die Produktion nach Störungen möglichst reibungslos weitergeführt bzw. wiederaufgenommen werden kann.

Aufgrund der Eigenschaften einer SOA ergeben sich neue Herausforderungen: Security-Ziele werden nicht nur innerhalb eines Unternehmens oder einer Sicherheitsdomäne angestrebt, sondern durchgängig für einen gesamten Geschäftsprozess und damit oft über mehrere Unternehmen hinweg.

- Schutz der Daten
  Sensible Daten, die im Rahmen eines Prozesses ausgetauscht werden, müssen bezüglich Integrität und Vertraulichkeit geschützt werden. Vertraulichkeit bedeutet, dass Daten nur von autorisierten Benutzern gelesen bzw. modifiziert werden dürfen. Dies gilt sowohl beim Zugriff auf gespeicherte Daten als auch bei der Datenübertragung. Integrität meint, dass alle Änderungen an Daten nachvollziehbar sein müssen und nicht unbemerkt durchgeführt werden dürfen. Zugriffe auf Daten müssen klar reglementiert werden. Dabei ist nicht nur zu berücksichtigen, wer auf die Daten zugreift, sondern auch in welchem Kontext diese Zugriffe erfolgen. So ergeben sich zum Beispiel bei einem Zugriff über das Internet spezifische Sicherheitsanforderungen.
  Des Weiteren muss die Verfügbarkeit der Daten garantiert werden, Systemausfälle sind zu verhindern. Zusätzlich muss die Bereitstellung der Daten innerhalb eines vereinbarten Zeitrahmens erfolgen. Nicht alle Daten unterliegen den gleichen Sicherheitsanforderungen. Eine Klassifizierung der Daten ist sinnvoll, um gezielte Maßnahmen für den jeweiligen Schutzbedarf (s. Kapitel 1.3) einzusetzen. Partner im Rahmen eines Geschäftsprozesses müssen sich über die jeweilige Schutzklasse der ausgetauschten Daten und über entsprechende Sicherheitsmaßnahmen einigen.
- Schutz von digitalen Identitäten
  Digitale Identitäten dienen dazu, Partner und Kunden im Geschäftsprozess zu identifizieren. Nicht nur fachliche Services und Anwendungen benötigen solche Identitätsinformationen. Es sind vor allem Sicherheitsfunktionen wie Authentifizierung und Autorisierung, die digitale Identitäten voraussetzen. Digitale Identitäten sind in der Regel personenbezogen und unterliegen damit den Datenschutzbestimmungen.
- Schutz der Services
  Wie für Daten sind auch für Services Integrität und Verfügbarkeit relevante Sicherheitsziele. Genau wie Daten können Services korrupt, gefälscht oder manipuliert sein. Deshalb gilt: Die Authentizität von Services, die als schützenswert eingestuft werden, muss vom Nutzer überprüfbar sein. Außerdem ist ein Service nutzlos, wenn er nicht zur Verfügung steht. Ein Service ist ebenso wie die Daten als schützenswert einzustufen, wenn er den Zugriff auf Daten ermöglicht, die als schützenswert klassifiziert sind.
- Nachvollziehbarkeit des Zugriffs
  Der Zugriff auf Daten und Services sowie Änderungen an denselben müssen nachvollziehbar und eindeutig einer Identität zuzuordnen sein. Dabei müssen Datenschutzbestimmungen beachtet werden: zum Beispiel bei der Protokollierung von erfolgten bzw. fehlgeschlagenen Zugriffen.
- Verbindlichkeit von (Geschäfts-)Transaktionen/Nichtabstreitbarkeit
  Häufig sind Transaktionen zwischen Geschäftspartnern verbindlicher Natur. Es ist notwendig, diese Verbindlichkeit durchzusetzen, damit der Vollzug einzelner Prozessschritte nicht abgestritten werden kann. Dazu müssen Nachweise erbracht werden können: z. B. darüber, ob eine Nachricht versendet oder empfangen worden ist. Darüber hinaus müssen die Urheber identifizierbar sein.
- Skalierbarkeit der Security
  Individuelle „Policies“ der Geschäftsprozesspartner sowie gegenseitige Policy-Abkommen fordern eine skalierbare Form der Security-Mechanismen. Wird beispielsweise für einen Geschäftspartner eine Smartcard-Authentifizierung gefordert, um bestimmte Prozessschritte ausführen zu können, so sollte ein bestehender passwortbasierter Log-in-Mechanismus dementsprechend für die jeweiligen Prozessschritte erweitert werden können.
- Nachvollziehbarkeit der Security
  Immer mehr Unternehmen müssen imstande sein, Fragen wie „Was darf welcher Benutzer wann machen?“ zu beantworten, um die Befolgung (Compliance) der zahlreichen rechtlichen Anforderungen nachweisen zu können. Sicherheitsbedarfe und Zugriffsreglementierungen müssen darum klar dokumentiert sein. Deren Einhaltung muss nachweis- und überprüfbar sein.

7.1.3 Sicherheitsbedarfe

Die Sicherheitsbedarfe in einer SOA müssen schon frühzeitig in die gesamte SOA-Planung einbezogen werden, sie müssen also auch in die Beschreibung der Geschäftsprozesse aufgenommen werden. Wie schon in Kapitel 1.2 beschrieben, ändern sich für eine SOA die Ziele und Prinzipien der Sicherheit nicht, sondern es stellen sich neue Herausforderungen an die Umsetzung von Sicherheitslösungen.

Hervorzuheben sind insbesondere:

- die Notwendigkeit, Identitäten von Nutzern und Diensten zu verwalten, insbesondere über Organisationsgrenzen hinweg
- die angemessene Umsetzung der Sicherheitsanforderungen für die Nutzung eines Dienstes und, bei einer Kombination von Diensten, das Erfassen und Überwachen des Schutzbedarfs jedes einzelnen Dienstes sowie des Schutzbedarfs des Geschäftsprozesses
- der Aufbau von Vertrauensverhältnissen zwischen Organisationen, die an einem Geschäftsprozess beteiligt sind
- die Entwicklung geeigneter Werkzeuge, die es wie MDA[1] erlauben, die Sicherheitsanforderungen in die Modellierung der Geschäftsprozesse aufzunehmen, die Security Services zu identifizieren und diese dann mittels Workflows umzusetzen

Der Schutzbedarf eines Geschäftsprozesses hat Auswirkungen auf jeden beteiligten Dienst, der zur Erfüllung des Geschäftsprozesses benötigt wird. Es ist sicherzustellen, dass der Schutzbedarf auf die einzelnen Dienste abgebildet wird. Jeder an einem Geschäftsprozess beteiligte Dienst muss mindestens den Sicherheitsanforderungen genügen, die sich aus dem Schutzbedarf des jeweiligen Geschäftsprozesses ergeben, da sich sonst Sicherheitslücken ergeben und schwach gesicherte Komponenten Angriffsziele darstellen. Für eine sichere Komposition sind daher Konzepte erforderlich, die es einem Dienstanbieter ermöglichen, den Nachweis zu erbringen, dass der Dienst den erforderlichen Schutzbedarf erfüllt. Dazu ist es notwendig, unterschiedliche Sicherheitslevel (Profile) zur Verfügung zu stellen, die entsprechend in die Dienste eingebunden werden können.

Neben der Beschreibung der Sicherheitsbedarfe ist es vor allen Dingen notwendig, die Einhaltung der daraus resultierenden Sicherheitsmaßnahmen zu überwachen. Dazu sind entsprechende Mechanismen wie Governance und Compliance bereitzustellen.

Security Governance kümmert sich um Strategie, Definition, Durchsetzung und Steuerung von organisatorischen Regeln, Richtlinien und Standards zur konsequenten Sicherung der Dienste und Geschäftsprozesse mittels geeigneter Steuerungs- und Kontrollmaßnahmen.

Security Compliance umfasst die Einhaltung von Verhaltensmaßregeln, Gesetzen und Richtlinien sowie Sicherheitseinstellungen und vergleicht sie mit den vorgegebenen Security Policies, was gegebenenfalls zur Korrektur von Abweichungen führt.

7.2 Geschäftsprozesse und deren Security-Anforderungen

7.2.1 Herausforderungen und Chancen

SOA und Geschäftsprozesse

Service-orientierte Architekturen können deutlich dazu beitragen, Geschäftsprozesse auf IT-Systeme abzubilden. Prozessschritte, die sich als Service-Aufrufe realisieren lassen, können in solchen Systemen leichter miteinander verbunden werden. Anstatt Anwendungen einzeln bedienen zu müssen, stoßen Geschäftspartner Prozesse an, die aus miteinander kommunizierenden Services bestehen. Standards spielen dabei eine zentrale Rolle, um die notwendige Interoperabilität herzustellen. Services können lokal im eigenen Unternehmen verfügbar sein oder von Geschäftpartnern und Serviceanbietern bereitgestellt werden. Die Flexibilität von SOA ermöglicht es, komplexe Geschäftsprozesse und die Beteiligung von unterschiedlichen Partnern und Serviceanbietern an diesen in der Software und den

darunterliegenden Systemen abzubilden.
Im Rahmen eines Geschäftsprozesses spielen die jeweiligen Sicherheitsrichtlinien der Geschäftspartner eine wichtige Rolle. Geschäftspartner schließen für gemeinsame Prozessschritte Abkommen über die Security Policy, um Services nutzen zu können, die von Partnern gehostet werden. Solche Policies regeln auch den Zugriff auf Daten. Notwendige Voraussetzung ist gegenseitiges Vertrauen hinsichtlich der angemessenen Umsetzung der vereinbarten Policy.

Herausforderungen

Service-orientierte Architekturen unterstützen flexible Geschäftsprozesse. Es ist jedoch aufwendiger als bei herkömmlichen Architekturen, die genannten Security-Ziele zu erreichen.

Der erhöhte Aufwand ist organisatorischer Natur, besonders dann, wenn Unternehmensgrenzen überschritten werden. Das Zusammenspiel verschiedener Geschäftsprozesspartner bedeutet letztlich ein Zusammenspiel von verschiedenen Security Policies. Anpassungen und Ergänzungen der jeweiligen Policies sind kritische Vorgänge, sie können weitreichende Konsequenzen haben.

Die Umsetzung einer Security Policy erfolgt innerhalb der Domänen. Dieser Vorgang ist zentral und geschäftsprozessübergreifend nur bedingt überprüfbar, besonders wenn verschiedene Partnerunternehmen sich am Geschäftsprozess beteiligen.

Der hohe Grad der Verteilung von Komponenten erhöht den Aufwand für die Absicherung der Kommunikation. Verteilte Komponenten sowie deren Umgebung müssen vor Angriffen von außen besonders geschützt werden. Über eine SOA werden heterogene Systeme miteinander verbunden. Es spielen unterschiedliche Produkte, Hersteller und Technologien zusammen. Für die Umsetzung der Policy müssen heterogene Sicherheitsmaßnahmen zusammenspielen. Eine Schwierigkeit besteht darin, dass diese unterschiedlichen Komponenten erst beim Service-Provider integriert werden können.

Chancen

Geschäftsprozesse und SOA bringen nicht nur neue Herausforderungen mit sich, sondern bieten auch Chancen für eine bessere Security: Durch eine SOA können logische Schichten eines Systems gut voneinander abgegrenzt und verteilt werden. Business-Logik, Datenzugriffslogik sowie Daten können durch Services gekapselt und durch die konsequente Umsetzung des SOA-Modells ausgelagert werden, um angemessen geschützt zu werden. Das Risiko von Angriffen innerhalb der eigenen Domäne wird oft unterschätzt – in der Tat können Insider den größten Schaden verursachen. Durch das Verteilen von Daten und Services in einer anderen Domäne, können Schutzmechanismen aufgestellt werden, die nicht mehr so leicht zu umgehen sind – bereits das Verhindern des physischen Zugriffs auf die Hardware kann das Risiko erheblich reduzieren. Die Verteilung von Daten und Services auf unterschiedliche Sicherheitsdomänen bietet die Möglichkeit für das effektive Durchsetzen von Aufgabentrennungen (Separation of Duties). Services sind nur durch klar definierte Schnittstellen zugänglich. Schnittstellen beschreiben einen Service, als sei dieser ein Kommunikationsendpunkt. Diese Kommunikationsendpunkte sind besonders gut geeignet, um Teilaspekte einer Security Policy durchzusetzen – sie dienen als Policy Enforcement Points.

7.2.2 Hauptmerkmale von Geschäftsprozessen im Rahmen von SOA und Security

Verteilung und Sicherheitsdomänen

Eine Sicherheitsdomäne ist die Menge von Daten, Identitäten und Services, für deren Sicherheit eine bestimmte Person oder Organisation verantwortlich ist. Identitäten sind typischerweise Benutzer, für die ein Benutzerkonto

in der Domäne eingerichtet ist. Ein Benutzerkonto kann als kleinste Einheit einer Sicherheitsdomäne betrachtet werden, die in der Verantwortung des jeweiligen Benutzers liegt.

Einzelne Prozessschritte werden entweder innerhalb einer Sicherheitsdomäne oder verteilt zwischen Domänen durchgeführt: Eine Service-orientierte Anwendungslandschaft unterstützt das Ausführen von verteilten Prozessschritten durch den Aufruf von Services unterschiedlicher Sicherheitsdomänen. Bereits eine SOA, die nur innerhalb eines Unternehmens eingesetzt wird, überschreitet Benutzerkonten und somit Domänengrenzen: Die Anwendung läuft nicht nur im Benutzerkonto, sondern verbindet sich mit Services, die im Kontext eines Serverkontos zur Verfügung stehen.

Vertrauensbeziehungen

Vertrauensbeziehungen zwischen den Sicherheitsdomänen der Beteiligten an einem Geschäftsprozess, sind die Grundlage für die Effektivität aller implementierten Sicherheitsmaßnahmen. Die Existenz einer Vertrauensbeziehung wird während des Geschäftsprozesses überprüft, indem sich die Partner im Prozess mittels einer digitalen Identität ausweisen und authentifizieren. Vertrauen heißt hier jedoch nicht, dass voller Zugriff auf Daten und Services erteilt wird, sondern in geregelter Form erfolgt, anhand von Richtlinien der vereinbarten Security Policy zwischen Geschäftspartnern.

Derartige Vertrauensbeziehungen können verschiedene Ausprägungen aufweisen. Zum Beispiel: Einem Mitarbeiter wird innerhalb der Domäne des eigenen Unternehmens vertraut. Er wird von den Geschäftspartnern des Unternehmens jedoch nicht automatisch als vertrauenswürdig eingestuft. Er kann sich aber typischerweise direkt gegenüber der Sicherheitsdomäne seines Unternehmens ausweisen (z. B. beim Log-in in das eigene Benutzerkonto innerhalb der Domäne).

Es gibt im Wesentlichen zwei Möglichkeiten:

- Für den Mitarbeiter kann eine direkte Vertrauensbeziehung zum Geschäftspartner aufgebaut werden. Dies geschieht typischerweise über ein Benutzerkonto in der Partner-Domäne, mit dem sich der Mitarbeiter gegenüber der Partner-Domäne direkt ausweisen kann.
- Es gibt auch Szenarien, in denen eine indirekte Vertrauensbeziehung genügt. Falls die vereinbarte Security Policy dies vorsieht, kann ein Mitarbeiter mit einem Benutzerkonto in einem Unternehmen in Verbindung mit einer bestehenden Vertrauensbeziehung auf Unternehmensebene von den Geschäftspartnern des Unternehmens als vertrauenswürdig eingestuft werden. Es handelt sich dabei um eine sogenannte Federation von Sicherheitsdomänen. Vertrauensintermediäre sind dann Sicherheitsdomänen bzw. Benutzerkonten, denen man beim Überprüfen einer indirekten Vertrauensbeziehung vertraut.

Die erhöhte Zusammenarbeit von Unternehmen im Rahmen einer SOA, die durch die Einbindung von Services unterschiedlicher Sicherheitsdomänen in einen Geschäftsprozess entsteht, macht indirekte Vertrauensbeziehungen besonders attraktiv. Federation ist im SOA-Umfeld fast zur Notwendigkeit geworden: Einerseits kann so der Administrationsaufwand beim Einrichten und Verwalten von Benutzerkonten in Grenzen gehalten werden, und andererseits kann dem Anwender und Partner im Geschäftsprozess ein Single Sign-on angeboten werden.

Vertrauensbeziehungen sind nicht nur Grundlage für eine Authentifizierung und Autorisierung, sondern auch für etliche andere Sicherheitsmaßnahmen im Rahmen von Geschäftsprozessen: beispielsweise für die Verteilung und das Management kryptografischer Schlüssel und Zertifikate für die Absicherung der Kommunikation zwischen Services sowie von End-to-End-Security-Mechanismen.

7.2.3 Typische Prozessklassen

Anhand der Merkmale Verteilung und Vertrauensbeziehung können folgende Szenarien unterschieden werden, die typisch für Prozesse im Rahmen einer SOA sind. Diese Szenarien können unterschiedliche Sicherheitsanforde-

rungen haben und zu unterschiedlichen Security Policies führen.

Prozesse innerhalb einer Sicherheitsdomäne

Es handelt sich hierbei um Aufrufe von Services innerhalb der Sicherheitsdomäne des Clients.

Verteilung: Diese Prozesse können die Grenzen einzelner Benutzerkonten überschreiten und in diesem Sinne verteilt sein. Client (C) und Service (S) liegen aber in Benutzerkonten einer gemeinsamen übergeordneten Sicherheitsdomäne.

Vertrauensbeziehung: Benutzerkonten werden innerhalb einer übergeordneten Domäne eingerichtet und administriert. Einem Benutzer, der sich über sein Konto authentifiziert, wird innerhalb der zuständigen Domäne direkt vertraut.

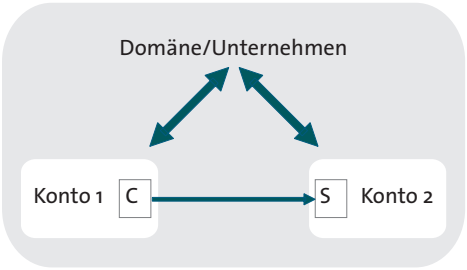


Abbildung 33: Schema Verteilung und Vertrauensbeziehung innerhalb einer Domäne

Prozesse zwischen Domänen, die sich „direkt“ vertrauen

Verteilung: Es werden Services außerhalb der Sicherheitsdomäne des Clients aufgerufen. Die Verteilung überschreitet nicht nur Benutzerkontengrenzen, sondern auch die jeweils übergeordnete Sicherheitsdomäne.

Vertrauensbeziehung: Der Aufrufer authentifiziert sich direkt an der Partner-Domäne. Die dafür benötigte digitale Identität ist dort bekannt bzw. als Benutzerkonto vorhanden.

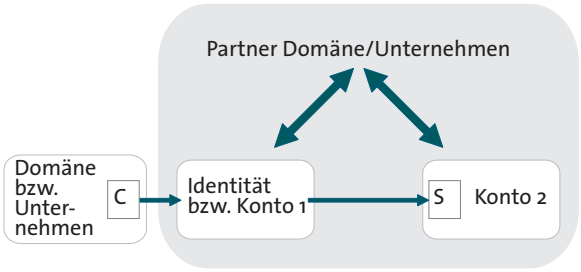


Abbildung 34: Schema Verteilung und direkte Vertrauensbeziehung zwischen zwei Domänen

Prozesse zwischen Domänen, in denen nur eine „indirekte“ Vertrauensbeziehung besteht

Verteilung: Es werden Services außerhalb der Sicherheitsdomäne des Clients aufgerufen. Die Verteilung überschreitet nicht nur Benutzerkontengrenzen, sondern auch die jeweils übergeordnete Sicherheitsdomäne.

Vertrauensbeziehung: Die Vertrauensbeziehung zwischen Aufrufer und Sicherheitsdomäne, in welcher der Service gehostet wird, wird über Vertrauensintermediäre hergestellt.

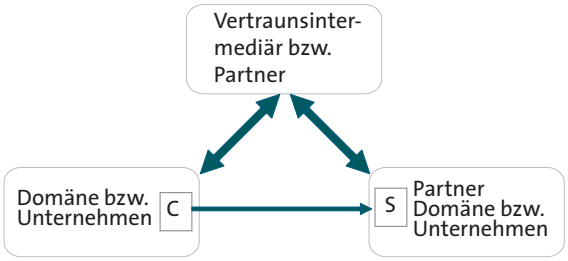


Abbildung 35: Schema Verteilung und Vertrauensverhältnis über einen Vertrauensintermediär

Weitere Prozesse

Komplexe Prozessabläufe bestehen aus Kombinationen der oben aufgeführten Arten von Prozessschritten.

7.2.4 Risiken und Sicherheitsanforderungen

Die typischen Bedrohungen für verteilte IT-Systeme sind bezüglich der aufgelisteten grundsätzlichen Security-Ziele (s. Abschnitt 1.2) weiterhin von Bedeutung, auch im Rahmen von SOA-unterstützten Geschäftsprozessen. Durch das Mitschneiden der Kommunikation zwischen Services oder durch unberechtigten Zugriff auf schützenswerte Services könnten zum Beispiel Daten ausspioniert und manipuliert werden. Authentifizierungsdaten könnten ebenfalls ausspioniert werden. Weitere Risiken sind denkbar.

Eine umfangreiche Risikoanalyse ist nicht Ziel dieses Artikels. Um Risiken zu minimieren, gibt es, je nach Prozessklasse, ganz bestimmte Sicherheitsanforderungen, die beachtet werden sollten. Diese sind in einer entsprechenden prozessübergreifenden Security Policy zu formulieren. Manche der Anforderungen betreffen die Architektur, insbesondere das Bereitstellen von Services im Rahmen der jeweils betrachteten Prozessklasse. Die folgende Liste der Sicherheitsanforderungen erhebt keinen Anspruch auf Vollständigkeit, vielmehr handelt es sich um eine Auswahl der aus unserer Sicht wichtigsten Anforderungen im Rahmen von SOA-unterstützten Geschäftsprozessen.

Anforderungen für Prozesse innerhalb einer Sicherheitsdomäne

Innerhalb einer Sicherheitsdomäne gibt es Einschränkungen bezüglich der erreichbaren Sicherheit. Daten, Services und Identitäten (Benutzerkonten) einer Sicherheitsdomäne sind vor den Administratoren der Domäne nicht geschützt. Darüber hinaus sind Daten und Services eines Benutzerkontos nicht vor dem Eigentümer des Kontos geschützt. Durch den Einsatz einer SOA ergeben sich trotzdem Chancen für eine angemessene Security, auch innerhalb einer Domäne.

Bereitstellung von schützenswerten Services und Daten

Um eine Beschränkung des Zugriffs und somit eine Trennung von Aufgabenbereichen effektiv durchzusetzen, sollte Folgendes beachtet werden:

- Daten und Services, auf die ein Benutzer nur beschränkten Zugriff erhalten soll, müssen auf ein anderes Benutzerkonto ausgelagert werden.
- Auf schützenswerte Daten außerhalb des eigenen Benutzerkontos darf nur über einen Service (bzw. Serverprogramm) zugegriffen werden. Dieser Service und die entsprechenden Daten liegen innerhalb desselben Benutzerkontos.

Authentifizierung

- Das Anmelden (Log-in) an einem Benutzerkonto der Domäne erfordert immer eine Authentifizierung des Benutzers.
- Das Authentifizierungsverfahren für das Log-in wird von der verantwortlichen Instanz der Sicherheitsdomäne definiert.
- Die Nutzung eines Service, der sich außerhalb des eigenen Benutzerkontos befindet (z. B. auf einem Server in der Domäne), erfordert eine Authentifizierung des Benutzers im Laufe des Geschäftsprozess, sofern der Service schützenswert ist.
- Das Authentifizierungsverfahren im laufenden Prozess wird von der verantwortlichen Instanz der Sicherheitsdomäne definiert (z. B. Kerberos).

Autorisierung

- Der Zugriff auf Daten, Services und Identitäten einer Sicherheitsdomäne muss autorisiert werden, sofern diese schützenswert sind. Die Identität des Subjekts in der Autorisierung muss zuvor authentifiziert werden.
- Die Autorisierung für Zugriffe auf Services und Daten, die in der Domäne eines Benutzers liegen, wird im Kontext desselben Benutzerkontos durchgesetzt.
- Einer Identität werden nur die notwendigen Berechtigungen erteilt, auf Services und Daten zuzugreifen, die im Rahmen einer bestimmten Tätigkeit (Rolle) erforderlich sind („least privilege“). Per Default werden einem Benutzer keine Berechtigungen zugeordnet (Compliance).

- Die Zuständigkeit für das Verwalten der Zugriffs-Policy sowie der Inhalt der Policy (d. h. „Wer darf was wie nutzen“) wird von der verantwortlichen Instanz der Sicherheitsdomäne definiert. Die Administration erfolgt in der Regel innerhalb der Sicherheitsdomäne.
- Die Zugriffs-Policy muss administriert werden können und sollte nicht statisch hinterlegt sein.

Kommunikation

Auch innerhalb einer Sicherheitsdomäne kann, besonders durch den Einsatz einer SOA, IP-basierte Kommunikation stattfinden, typischerweise zwischen Benutzerkonten oder Rechnern. Diese Form der Kommunikation muss abgesichert werden, zum Beispiel mittels Mechanismen wie Kerberos (siehe Liste oben). Da die Kommunikation innerhalb einer Sicherheitsdomäne stattfindet, ist das Absichern derselben durch die vorhandene direkte Vertrauensbeziehung der Benutzer der Domäne erleichtert.

Protokollierung

Wird ein Schritt eines Geschäftsprozesses im Namen einer Identität ausgeführt, die von der des ursprünglichen Benutzers abweicht, muss es möglich sein, diese ursprüngliche Identität nachzuvollziehen. Innerhalb einer Sicherheitsdomäne ist dies zum Beispiel der Fall, wenn ein Funktionsbenutzer, unter dem ein Service läuft, auf Daten zugreift. Es muss außerdem möglich sein, nachzuvollziehen, auf welche Daten und Services mit welcher Identität zugegriffen wurde.

- Sowohl beim Log-in an einem Benutzerkonto als auch beim Zugriff auf schützenswerte Daten und Services muss ein entsprechendes Ereignis (Event) registriert werden können. Jedes Ereignis muss einer Identität/einem Benutzer eindeutig zuzuordnen sein.
- Die Protokollierung muss gemäß der vorhandenen Policy identitätsspezifisch ein- und abschaltbar sein (Konformität zu Datenschutzbestimmungen).

Anforderungen für Prozessschritte zwischen Domänen, die sich „direkt“ vertrauen

Authentifizierung

- Die Nutzung eines Service außerhalb seiner Sicherheitsdomäne erfordert die Authentifizierung des Benutzers oder der Partner-Sicherheitsdomäne, sofern der Service schützenswert ist.
- Der Benutzer ist in der Sicherheitsdomäne des Service-Providers bekannt und kann von dieser direkt authentifiziert werden. Dafür müssen sich Benutzer beim Service-Provider registrieren oder es müssen entsprechende Konten in der Service-Provider-Domäne angelegt werden.
- Das Authentifizierungsverfahren für den Service-Aufruf, d. h. für die direkte Authentifizierung des Benutzers, wird von der verantwortlichen Instanz der Sicherheitsdomäne des Service definiert. Dieses Verfahren könnte aber auch im Einverständnis von Benutzer bzw. Unternehmen und Service-Provider definiert werden.
- Für die Authentifizierung zwischen Sicherheitsdomänen sind Standardprotokolle einzusetzen, um eine möglichst hohe Plattformunabhängigkeit zu erreichen. Für die Bereitstellung von Authentifizierungstokens müssen Standardstrukturen wie z. B. WS-Security und SAML verwendet werden.
- Authentifizierungstokens müssen genügend Merkmale aufweisen, um die entsprechenden Benutzer gegenüber der Service-Provider-Domäne zu identifizieren und um die direkte Vertrauensbeziehung zu beweisen.

Autorisierung

Zusätzlich zu den Forderungen für Prozessschritte innerhalb einer Sicherheitsdomäne ist Folgendes zu beachten:

- Die Autorisierung für Zugriffe auf Services und Daten der Provider-Domäne wird im Kontext der Provider-Domäne durchgesetzt.
- Der Inhalt der Zugriffs-Policy („Wer darf was wie nutzen“) wird in der Regel im Einverständnis von Benutzer bzw. Unternehmen und Service-Provider festgelegt.

Kommunikation

Es findet in der Regel eine IP-basierte Kommunikation zwischen Benutzer und Service in der Provider-Domäne statt, um SOAP-Nachrichten auszutauschen.

- Falls schützenswerte Daten übertragen werden, sollte ein sicherer Transportkanal aufgebaut werden (z. B. über SSL, IPSec usw.).
- Schützenswerte Daten, die zwischen Sicherheitsdomänen ausgetauscht werden, sollten außerdem auf Nachrichtenebene (Message Security) geschützt werden, um den Schutz von der aufrufenden Anwendung bis zum Service – „end-to-end" – herzustellen. Dafür sollten die entsprechenden Standards eingesetzt werden (XML-Security, WS-Security, ...).
- Die Sicherheitsstufe für Transportkanäle und Nachrichten wird von der verantwortlichen Instanz der Provider-Domäne festgelegt, kann aber auch im Einverständnis von Benutzer bzw. Unternehmen und Service-Provider definiert werden.

Protokollierung

- Sowohl bei der Authentifizierung eines Benutzers und der Service-Provider-Domäne als auch beim Zugriff auf Funktionen eines schützenswerten Service muss ein entsprechendes Ereignis (Event) registriert werden können. Jedes Ereignis muss einer Identität/einem Benutzer eindeutig zuzuordnen sein.
- Die Protokollierung muss gemäß der vorhandenen Policy identitätsspezifisch ein- und abschaltbar sein (Konformität zu Datenschutzbestimmungen).

Anforderungen für Prozessschritte zwischen Domänen, die sich „indirekt" vertrauen

In diesem Szenario sind besondere Anforderungen bezüglich Authentifizierung sowie zur Autorisierung und Protokollierung von Zugriffen auf Ressourcen zu beachten. Die Anforderungen zur Sicherheit der Kommunikation sind im Wesentlichen dieselben wie in den vorherigen Szenarien.

Authentifizierung

- Die Nutzung eines Service durch einen Client außerhalb der Sicherheitsdomäne des Service-Providers erfordert eine Authentifizierung, sofern der Service schützenswert ist.
- Damit der Service-Provider den Aufrufer als authentifizierte Instanz akzeptiert, ist eine Vertrauensbeziehung zum Service-Provider erforderlich. In dieser Vertrauensbeziehung werden Vertrauensintermediäre gestattet. Es ist daher nicht erforderlich, dass der Aufrufer der Sicherheitsdomäne des Service-Providers direkt bekannt bzw. als Benutzer beim Service-Provider registriert ist. Dem Benutzer wird auf Basis der Vertrauensintermediäre (Federation) vertraut.
- Der Service-Provider muss bei der Definition der Security Policy entscheiden, ob Vertrauensintermediäre gestattet sind. Das Authentifizierungsverfahren muss vom Service-Provider und von den betroffenen Vertrauensintermediären unterstützt werden.
- Authentifizierungstokens müssen genügend Merkmale enthalten, um die direkte Vertrauensbeziehung zu einem Vertrauensintermediär beweisen zu können. Außerdem sollte die Identität, Pseudonym oder Stellvertreter des ursprünglichen Benutzers sowie aller Vertrauensintermediäre im Token enthalten sein. Wird ein Schritt im Geschäftsprozess im Namen einer anderen Identität ausgeführt, die von der des ursprünglichen Benutzers abweicht, muss es möglich sein, diese ursprüngliche Identität nachzuvollziehen.
- Für die Authentifizierung sind Standardprotokolle und Standardstrukturen einzusetzen. Dies ist besonders wichtig, da die Authentifizierungs-Policy nicht bilateral zwischen Benutzer und Service-Provider definiert wird, sondern mehrere Partner betrifft.

Autorisierung

- Die Autorisierung des Zugriffs muss anhand der Identität (oder eines Pseudonyms) des ursprünglichen Benutzers erfolgen, um die Nachvollziehbarkeit des Zugriffs während des gesamten Geschäftsprozesses herstellen zu können.
- Der Inhalt der Zugriffs-Policy wird vom Service-Provider festgelegt, ggf. in Abstimmung mit Vertrauensintermediären. Die End-Benutzer sind typischerweise nicht involviert, da keine direkte Vertrauensbeziehung zwischen Benutzer und Service-Provider besteht.

Protokollierung

- An allen Stellen, an denen ein Identitätswechsel stattfindet, muss ein entsprechendes Ereignis registriert werden.

Allgemeine Usability-Anforderungen
Ein Single Sign-on ist für die Gesamtheit eines Geschäftsprozesses anzustreben – ein Benutzer wird sich aber aus Sicherheitsgründen für bestimmte Services im Gesamtgeschäftsprozess trotzdem erneut authentifizieren müssen.

7.2.5 Sicherheitsanforderungen, Policy und Architektur

Aus den erwähnten Security-Anforderungen lassen sich für die Teilnehmer eines Geschäftsprozesses wichtige Bausteine einer übergreifenden Security Policy ableiten. Um solche Policies zu administrieren und durchzusetzen, muss die eingesetzte SOA entsprechend ausgerichtet sein. Von den aufgelisteten Anforderungen betreffen einige die Security-Architektur der Services. Diese Anforderungen können durch den Einsatz sogenannter Policy-Systeme umgesetzt werden. Die wichtigsten Aspekte und Bausteine eines Policy-Systems werden in den folgenden Absätzen erläutert. Die Bausteine eines Policy-Systems lassen sich in eine SOA gut integrieren. Es sind insbesondere die Policy Enforcement Points, die an mehreren Stellen einer SOA aufgestellt werden müssen, um die Security im Geschäftsprozess zu verankern. Das Policy-System an sich existiert parallel zu den eigentlichen fachlichen Services und Geschäftsprozessen. Dabei könnte das Policy-System selber die Eigenschaft haben, Service-orientiert zu sein, im Sinne von „Security as a Service“ mit den Enforcement Points, die als Clients dienen (Abbildung 4).

7.3 Policy Management

7.3.1 Architekturbetrachtungen

Durch die Einführung von SOA-Konzepten wird das Modell logisch und physisch verteilter Anwendungen zur gängigen Praxis. Die damit verbundene Intention, Softwarekomponenten ohne großen Aufwand wiederzuverwenden, impliziert unter anderem, dass es möglich sein muss, Komponenten zu einer logischen Anwendung zusammenzufügen, die mit unterschiedlichen Techniken entwickelt wurden, von unterschiedlichen Herstellern stammen und die jeweils eigene Methoden haben, Anforderungen an ihr Verhalten zu spezifizieren und umzusetzen.

Es ist notwendig, über ein solches verteiltes System, einer Klammer gleich, ein Netz von organisationsweit einheitlichen Richtlinien zu legen. Dabei ergibt sich zwangsläufig eine Hierarchie.

- An der Spitze stehen Anforderungen, die sich aus den jeweiligen geschäftlichen oder rechtlichen Erfordernissen ergeben und die in natürlicher Sprache formuliert werden. Sie können generischer Natur sein – Beispiele: Alle Daten müssen gemäß ihrer Klassifizierung vor unberechtigter Einsichtnahme durch geeignete Maßnahmen geschützt werden. Jeder Zugriff auf die zentrale Personaldatenbank muss protokolliert werden. Was nicht ausdrücklich erlaubt ist, ist

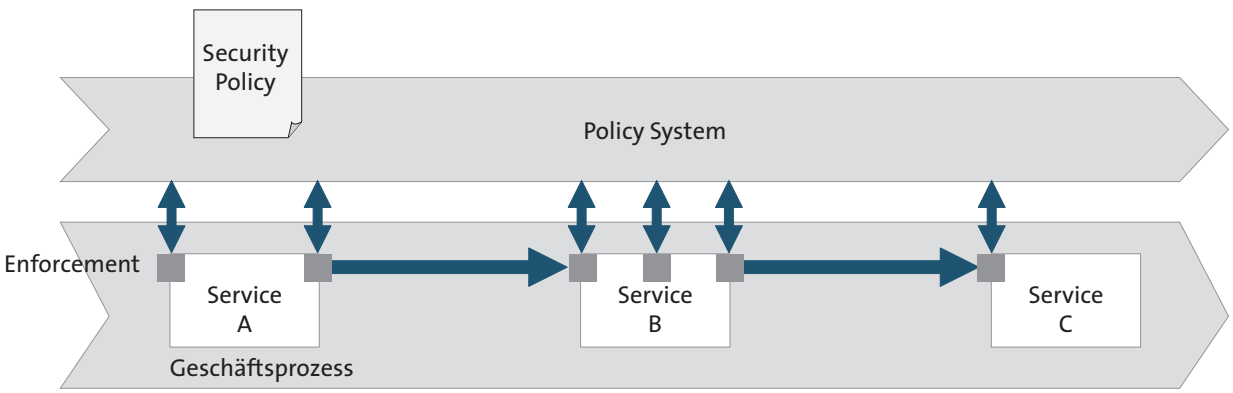


Abbildung 36: Schema Geschäftsprozesse und Policy-System

verboten. Auch das umgekehrte Prinzip gibt es: Was nicht explizit verboten wird, ist erlaubt. Das Need-to-know-Prinzip und das Vier-Augen-Prinzip sind weitere mögliche Ansätze.

- Hieraus folgend werden einige Konkretisierungen notwendig, zum Beispiel: ein Klassifizierungsschema für Daten; Kriterien, aus denen Benutzerrechte abgeleitet werden können (z. B. Gruppen, Rollen, Lokation, Mitarbeiterstatus); Authentifizierungsmethoden; Aussagen über personenbezogene Daten; Aussagen zu Außenbeziehungen (externe Kommunikation)
- Am unteren Ende steht eine Umsetzung der Anforderungen in den IT-Systemen. Die Verteiltheit und Vielfalt von SOA bedeutet somit, dass die Vorgaben (Policies) an vielen Punkten im Gesamtsystem vorgehalten, evaluiert und umgesetzt werden müssen.

Man sieht deutlich, dass es an dieser Stelle einen Bruch gibt, der den Bedarf an einem zentralen, einheitlichen Policy Management aufzeigt, das als Bindeglied wirkt. Damit die Vielfalt handhabbar bleibt, ist es erforderlich, ein solches Management mit einheitlichen Standards auszustatten. Ein solcher Standard ist XACML (OASIS: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf). Er beschreibt sowohl ein Architekturmodell als auch eine Sprache, in der Policies formuliert werden und Autorisierungsentscheidungen abgefragt werden können. Dabei soll das Szenario, auf das sich die Policies im Einzelnen beziehen, der in der folgenden Grafik skizzierten Abstraktion entsprechen. Ein Subjekt (auch Requestor genannt) will auf eine Ressource (auch Objekt oder Target genannt) zugreifen, um eine bestimmte Aktion daran vorzunehmen. Der Zugriff wird von einer Zwischenkomponente kontrolliert, die entscheiden kann, ob sie den Zugriff zulässt, abweist oder mit bestimmten Bedingungen verknüpft, die der Requestor einzuhalten hat.

Im Folgenden wird am Beispiel des XACML-Modells dargestellt, was die wichtigsten Eigenschaften eines Policy-Systems sind. Die durch ein solches System zu erfüllenden Aufgaben sind: schreiben, ansehen, testen, freigeben, verteilen, kombinieren, analysieren, verändern, löschen und durchsetzen von Policies. XML ist leicht zu erweitern und bereits weitverbreitet, es ist daher die ideale Basis für eine Security-Policy-Sprache.

Aufgabenteilung, Basiskomponenten

Das folgende Bild stellt die vier elementaren Funktionen in einem Policy-System dar sowie die sich daraus ergebenden Basiskomponenten und ihr Zusammenspiel. Dabei stellt die Administrationskomponente (PAP) im Zusammenspiel mit dem Policy Decision Point (PDP) das im vorigen Abschnitt gesuchte Bindeglied zwischen Abstraktion und konkreter Umsetzung dar. In den folgenden Abschnitten werden die Komponenten im Einzelnen vorgestellt.

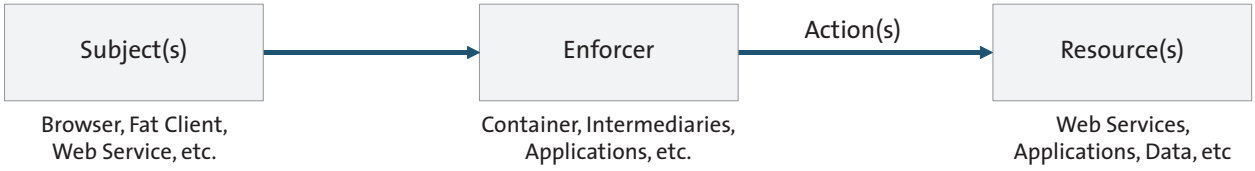


Abbildung 37: Szenario eines Policy Enforcement

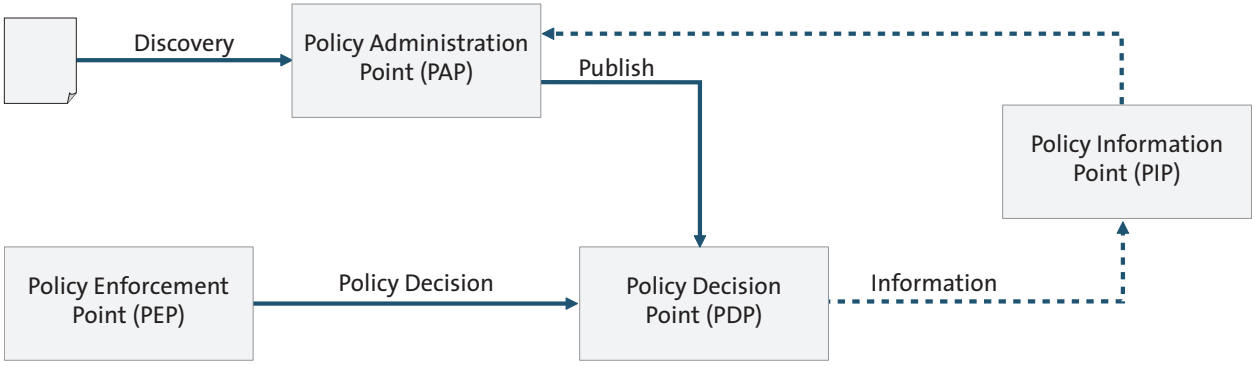


Abbildung 38: Basiskomponenten eines Policy-Systems

Policy Administration Point (PAP)

Die Erstellung und Modifikation von Policies erfolgt zunächst im Policy Administration Point in abstrakter Form. Was ist damit gemeint? Der Kontext einer Policy-Umsetzung, d. h. einer Autorisierungsentscheidung, bedarf zunächst einer Abstraktion der in Abbildung 5 dargestellten Operation, dem Entscheidungskontext (Subjekt, Ressource, Aktion, Environment). Ferner gilt es, den Typ der Policy festzulegen (z. B. Autorisierungs-Policy, Message Protection Policy, Auditing Policy). Im Falle der Autorisierungs-Policy wird die Aktion entweder erlaubt (permit) oder untersagt (deny). Dies geschieht in Abhängigkeit von bestimmten Bedingungen: Gruppen- bzw. Rollenzugehörigkeit des Subjekts, Bedingungen für bestimmte Attribute von beispielsweise Subjekt oder Ressource. Diese Bedingungen werden abstrakt definiert und mittels logischer Operatoren (und/oder) miteinander verknüpft.

Um aus der abstrakten eine konkrete Policy zu machen, müssen die Elemente aus dem Kontext mit realen Entitäten verknüpft (gebunden) werden, z. B. wird angegeben, dass man die Attribute des Subjekts in einem bestimmten LDAP-Verzeichnis findet. Ein weiteres Beispiel: Wenn die Ressource ein Webservice bzw. eine Funktion eines Service darstellt, kann dieser durch den Import seiner Meta-Daten (WSDL) spezifiziert werden.

Ein Bestandteil einer solchen Policy kann die Auflage sein, dass zum Zugriffszeitpunkt bestimmte dynamische Daten hinzugezogen und evaluiert werden müssen. In diesem Fall erfolgt die Verknüpfung mit realen Objekten an anderer Stelle, nämlich im Policy Information Point.

Schließlich gehört es zu den Aufgaben des PAP, die erstellten Policies überall dorthin zu verteilen, wo sie im Kontext eines konkreten Zugriffes evaluiert werden müssen. Dabei kann es notwendig sein, Policies aus einem Standardformat wie XACML in ein anderes etwa herstellerspezifisches Format eines Access Control Systems gleichwertig umzuwandeln. Die Evaluierung kann dann auf Basis dieses proprietären Formates erfolgen.

Policy Decision Point (PDP)

Die zentrale Aufgabe des PDP besteht darin, zum Zeitpunkt eines Zugriffes die für die Evaluierung der geltenden Policy relevanten Daten zusammenzutragen und mit deren Hilfe die Policy zu evaluieren und somit eine Entscheidung zu treffen. Dazu muss der PDP die Policies vom PAP entgegennehmen und speichern. Der PDP wird durch einen Request vom PEP angestoßen und gibt die Entscheidung in einer Response zurück.

Die erste Aufgabe besteht darin, die für den Request geltende und zu evaluierende Policy (oder mehrere) zu ermitteln.

Die relevanten Daten stehen im Kontext, der Bestandteil des Requests ist. Werden zusätzliche Attribute benötigt (dies ergibt sich aus der Policy), müssen diese mit Hilfe des PIP ermittelt werden.

In Abhängigkeit von der jeweiligen Umgebung kann ein PDP mit einem PEP zusammenfallen. Dennoch handelt es sich stets um zwei logisch voneinander getrennte Funktionen.

Policy Information Point (PIP)

Beim PIP handelt es sich um einen Attributservice, der sowohl vom PDP als auch vom PEP befragt werden kann. Attribute können sich auf das Subjekt, die Ressource oder das Environment beziehen, also auf den gesamten Entscheidungskontext. Hierbei handelt es sich um einen sehr allgemeinen und damit vielseitigen Ansatz. Der XACML-Standard beschränkt sich auf eine allgemeine Darstellung dieser Funktionalität, vermeidet aber einschränkende Spezifikationen.

Policy Enforcement Point (PEP)

Der PEP ist am stärksten abhängig von der konkreten Laufzeitumgebung. Er kann in vielfältigen Ausprägungen und Platzierungen vorkommen. Um einen Zugriff über einen externen PEP zu erzwingen (d. h. einen PEP, der weder in den Requestor noch in das Target integriert ist), kann es erforderlich sein, im Netz entsprechende Maßnahmen zu treffen (Firewalls, Screening Router). Eine weitere Möglichkeit besteht darin, das Target so einzurichten, dass Requests ausschließlich von den vorhandenen PEPs entgegengenommen werden. Dies kann beispielsweise durch Prüfung von Zertifikaten oder speziell für diesen Zweck eingerichteten technischen Usern erreicht werden.

Beispiele

- Ein HTTP Reverse Proxy in der DMZ, der dem eigentlichen Webserver vorgeschaltet ist und in transparenter Weise Kontrolle über jeden Request der Browser–Webserver-Kommunikation bekommt, kann für diese Requests PEP sein.
- Der Anwendungscontainer auf einem Applikationsserver, der die Requests an die Anwendungen weiterleitet, kann gleichzeitig als PEP agieren.
- Message-Queuing-Systeme, deren Aufgabe darin besteht, die Übermittlung einer Nachricht zuverlässig zu garantieren, werden besonders häufig bei unternehmenskritischen Anwendungen mit sensitiven Daten verwendet. An den Ein- und Ausgängen dieser Systeme kann ein PEP platziert werden, der jedes Senden und Empfangen von Nachrichten autorisiert.
- Webservice Proxy als „Spiegel“ für die angebotenen Webservice-Schnittstellen, die ausschließlich auf diesem Weg erreichbar sein dürfen. Häufig findet man solche Funktionalitäten zusammen mit hardwarebasierter XML-Beschleunigung, d. h. auf speziellen Appliances.
- Enterprise Service Bus (ESB) Plug-in für SOA-Umgebungen – hier spielt der ESB eine entsprechende Proxy-Rolle. Das Plug-in wird mit den Mitteln der jeweiligen ESB-Plattform in die dort definierten Workflows geeignet und transparent eingebettet und übt die PEP-Funktionalität aus ohne den Nachrichtenfluss zu verändern.
- Kernel Interceptor für Betriebssystem-Funktionsaufrufe – hierbei handelt es sich um eine tiefe Integration, die stark von der Architektur des jeweiligen Betriebssystems abhängt. Jeder Aufruf von Betriebssystemfunktionen (z. B. Lesen und Schreiben von Dateien) wird abgefangen und zunächst auf Berechtigung geprüft.

Da eine Autorisierungsentscheidung als Resultat einer Policy-Evaluierung auf einem PDP nicht auf eine Ja-/Nein-Antwort beschränkt sein muss, sondern mit zusätzlichen Auflagen verknüpft sein kann (in XACML = Obligations), muss der PEP die entsprechenden Fähigkeiten besitzen, diese Auflagen zu erfüllen. Zum Beispiel kann eine Policy vorschreiben, dass Daten verschlüsselt werden müssen. In diesem Fall kann es erforderlich werden, dass der PEP hardwarebasierte Kryptographiebeschleuniger unterstützt.

Bei einem verteilten Policy-System wird die Kommunikation zwischen den einzelnen Komponenten selbst zum

Gegenstand von Security-Betrachtungen. Der Basis-XACML-Standard beschreibt nicht, wie die Kommunikation zwischen PEP und PDP gegen Angriffe zu sichern ist. Ebenso wenig wird die Art der Implementierung vorgegeben. Erst solche Vorgaben erlauben es, die hier aufgezeigte Vielfalt von Anwendungsszenarien anzugehen. Es liegt nahe, die fehlenden Vorgaben mit anderen bereits vorhandenen Standards etwa für Transportprotokolle, Nachrichtenverschlüsselung und -signierung zu schließen. XACML macht eine Ausnahme: Es gibt ein spezifiziertes Profil für die Anwendung von XACML in Verbindung mit SAML-2.0-Nachrichen und dem -Protokoll, wobei SAML seinerseits wieder andere Standards einbindet. Man erhält definierte Mechanismen für die Verschlüsselung und die digitale Signatur von ganzen Nachrichten oder einzelnen Nachrichtenelementen, für die Authentifizierung von Anfragen, die Kontrolle der Gültigkeitsdauern von Nachrichten und Nachrichtenelementen usw.

7.3.2 Relevante Standards

WS-Policy

WS-Policy beschreibt in abstrakter Form die Charakteristiken eines Webservice:

- Anforderungen an den Client
- Fähigkeiten, die der Service hat

WS-Policy besteht aus einzelnen Assertions, dies sind nicht mehr weiter in Bestandteile zerlegbare Anforderungen. Es gibt zwingende (mandatory) und optionale Assertions.

Die Zuordnung zu einem Service geschieht mittels WSDL Binding, also statisch über die Servicebeschreibung oder mittels UDDI Referencing, also dynamisch über ein Serviceverzeichnis. Dies ist nicht Bestandteil der WS-Policy-Spezifikation.

WS-Policy beschreibt nur eine Syntax, in der Policy Assertions zu einer Policy zusammengesetzt werden. Die inhaltliche Interpretation der einzelnen Policy Assertions bleibt dabei anderen Standards überlassen: WS-Security Policy, WS-Reliable Messaging Policy, WS-Atomic Transaction usw.

Kombinationsmöglichkeiten sind z. B. die Auswahl genau einer aus mehreren Policies und Verschachtelung (Nesting) von Policies.

WS-Security Policy

WS-Security Policy beschreibt Policy Assertions (Syntax und Semantik), die sich auf die Anwendung von WS-Security, WS-Trust und WS-Secure Conversation beziehen.

Es gibt folgende Kategorien von Assertions:

- Protection Assertions, unterschieden nach Integrity (Signatur) und Confidentiality (Verschlüsselung)
- Token Assertions, sie stellen das geforderte Tokenformat sicher (z. B. Username Token, X509 Token, Kerberos Token, SAML Token) und ggf. Attribute (Claims)

Security Binding Assertions beschreiben, mit welcher kryptografischen Methode die Message mit dem Token zu verknüpfen ist: Transport Binding (z. B. HTTPS), Symmetric Binding (SOAP Message Security mit symmetrischen Schlüsseln), Asymmetric Binding (SOAP Message Security mit Public-Key-Technologie)

Supporting Token Assertions stellen (bei Bedarf) Anforderungen an weitere Token, mit denen zusätzliche Claims belegt werden sollen.

XACML

XACML dient dem Zweck, zu einer Autorisierungsentscheidung zu kommen. Das Modell ist die kontextbasierte (bzw. attributsbasierte) Autorisierung. Der Standard beschreibt sowohl die Policy (XML-Schema) als auch die Request-/Response-Nachrichten.

Es handelt sich um eine Verallgemeinerung sowohl der Role-based Access Control (z. B. in J2EE) als auch der ACL-based Access Control (z. B. in UNIX).

Ein Decision Request, also die Anforderung einer Autorisierungsentscheidung, enthält mindestens drei Informationen über die Art der zu autorisierenden Operation: Subjekt (wer), Ressource (was) und Aktion (wie). Optional können weitere Informationen enthalten sein (Environment).

Die Spezifikation beschränkt sich auf den Kern dessen, was bei der Entscheidungsfindung passiert. Sie beschreibt drei XML-Formate: den Request-Kontext, die Policy sowie die Response. Sie beschreibt nicht, wie die Informationen aus der jeweiligen Anwendungsumgebung (z. B. J2SE, COBA) in diese Formate umzuwandeln sind. Dies muss in den Implementierungen erfolgen.

Policies bestehen aus Regel-Elementen (Rules), die mittels eines Algorithmus (z. B. „Deny gewinnt", „Permit gewinnt") zu Policy-Elementen zusammengesetzt werden. Diese können ihrerseits durch weitere Rules zu Policy-Set-Elementen kombiniert werden. Ein weiteres (optionales) Element in einer Policy ist die Obligation, d. h. eine Auflage, die im Rahmen der Autorisierung erteilt wird. Nur wenn die Auflage erfüllt ist, gilt die Autorisierung. An dieser Stelle gibt es eine Überlappung mit WS-Security Policy: Die Security Policy als Eigenschaft eines spezifischen Service ist in dessen Beschreibung enthalten und dient somit einem Client dazu, den Service Request geeignet aufzubauen. Sie alleine stellt ihre eigene Einhaltung nicht sicher. Fehlerhafte oder gar böswillige Client-Implementierungen bedürfen zusätzlich der Überwachung. Hierzu dient die XACML-Obligation, sie wird von einem Policy Enforcement Point verwendet, um zu verifizieren, dass die Vorgaben tatsächlich zum Zugriffszeitpunkt erfüllt sind.

SAML im Kontext von XACML

XACML beschreibt ein SAML-Profil, in dem dargestellt wird, wie sich XACML-Requests und -Statements (Responses) mit Mitteln der SAML-Standards realisieren lassen. Dabei wird einerseits auf bestehende Formate zurückgegriffen (Attribute Query, Attribute Statement), andererseits werden auch Erweiterungen von SAML-Formaten definiert, um XACML Policy-Abfragen und Abfragen von Autorisierungsentscheidungen abzubilden.

7.3.3 Policy Definition und Policy Enforcement

Das Policy Management untergliedert sich in die drei wichtigen Handlungsebenen Policy Definition, Policy Enforcement und Policy Monitoring. Die Policy Definition umfasst die Erstellung der Richtlinien entsprechend den geschäftlichen Anforderungen sowie deren Aufbereitung für die Übergabe an die Handlungsebene des Policy Enforcements. Das Policy Enforcement ist dagegen eine reine IT-getriebene Phase, in der die Richtlinien in den Applikationen, Zwischenstationen oder Diensten durchgesetzt werden. Damit liegen die entscheidenden Mechanismen für einen Brückenschlag zwischen den geschäftlichen Anforderungen und einer flexiblen Umsetzung in

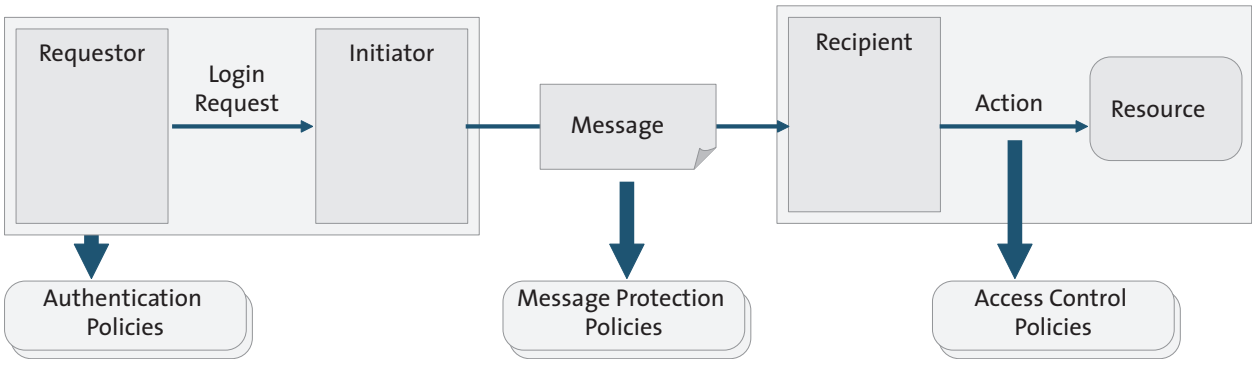


Abbildung 39: Policy-Szenario

der IT überwiegend in diesen zwei Handlungsebenen. Das Policy Monitoring liefert zusätzlich die Möglichkeit, die Durchsetzung und die Wirksamkeit der Schutzmaßnahmen auf Basis der Richtlinien zu überprüfen. Die Ergebnisse dieser Phase können und sollten in der Erstellung bzw. Anpassung der Richtlinien einfließen.

Für eine erfolgreiche Unternehmensführung ist ein integrierter Ansatz bzgl. Governance, Risk und Compliance (GRC) notwendig. Governance beinhaltet u. a. die Vorgabe unternehmensweiter strategischer Richtlinien. Im Risikomanagement können auf Basis spezieller Methoden frühzeitig Bedrohungsszenarien identifiziert und entsprechende Maßnahmen in Form von Richtlinien abgeleitet werden. Compliance umfasst die Überwachung der Richtlinien in Bezug auf ihre Einhaltung im operativen Betrieb. Das Policy Management entspricht den Grundsätzen des GRC-Modells. Dabei werden die Themen Governance und Risikomanagement vorwiegend in der Handlungsebene der Policy Definition behandelt, das Thema Compliance dagegen in den Ebenen von Policy Enforcement und Policy Monitoring.

Policy Definition

Die Erstellung von Richtlinien basiert auf einem inkrementellen Ansatz, der üblicherweise mit der Definition von allgemeinen unternehmensrelevanten Richtlinien (Corporate Policies) startet. Von diesen werden nach einem geeigneten Klassifizierungsschema weitere Richtlinien (z. B. Richtlinien untergliedert nach Prozessklassen) abgeleitet und mit zusätzlichen Informationen verfeinert. Im Rahmen der Verfeinerung in einzelnen Schritten sollten die Richtlinien dann in Gruppen gegliedert werden, die den Sicherheitsverfahren Nachrichtenschutz (Message Protection), Authentifizierung und Autorisierung (Access Control) zuzuordnen sind. Diese Gruppierungen sind in den unterschiedlichen Technologien und Standards begründet, die für die jeweiligen Sicherheitsverfahren bezüglich Policy Management anwendbar sind: WS-Security Policy für den Nachrichtenschutz, SAML für die Authentifizierung und XACML für die Autorisierung. Am Ende der Hierarchie (siehe Abbildung 9) müssen Richtlinien zur Verfügung stehen, die von den jeweiligen IT-Systemen interpretiert und angewendet werden können.

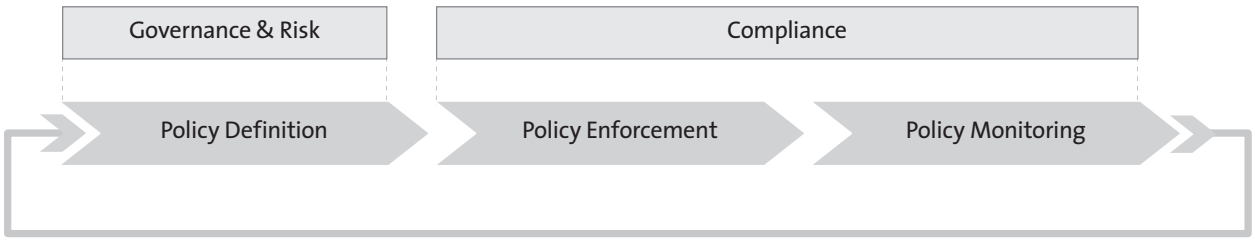


Abbildung 40: Handlungsebenen Policy Management

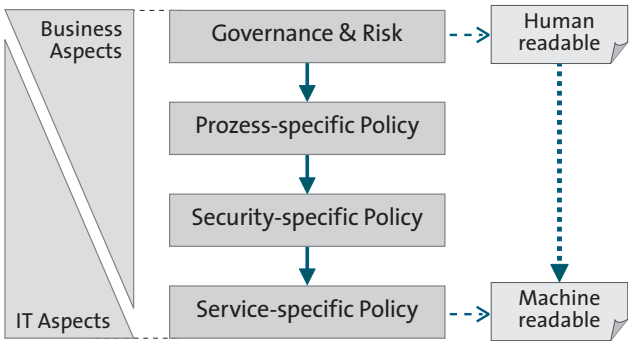


Abbildung 41: Policy-Hierarchie

Abgesehen von der inkrementellen Verfeinerung durchlaufen die Richtlinien in der Hierarchie von Ebene zu Ebene einen Transformationsprozess. Liegen die Richtlinien auf der oberen Ebene noch in einer frei gestalteten, allgemeinen und menschenlesbaren Form vor (z. B. Richtlinien-Dokument), so müssen die Richtlinien spätestens auf der untersten Ebene in einer standardisierten, konkreten und maschinenlesbaren Form (z. B. XML) den IT-Systemen übergeben werden. Die Kernfrage hierbei ist, ob und in welcher Ebene man den Transformationsprozess automatisieren kann. Wurden in den Zwischenebenen bereits (standardisierte) Modelle wie BPMN oder UML verwendet, ist eine automatische Transformation in die nachfolgende Ebene technologisch realisierbar. Die Kluft zwischen der geschäftlichen und der IT-Sicht wird an den Stellen besonders spürbar, wo noch manuelle Übersetzungen notwendig sind. Je größer der Automatisierungsgrad ist, desto geringer wird diese Kluft in der Praxis ausfallen.

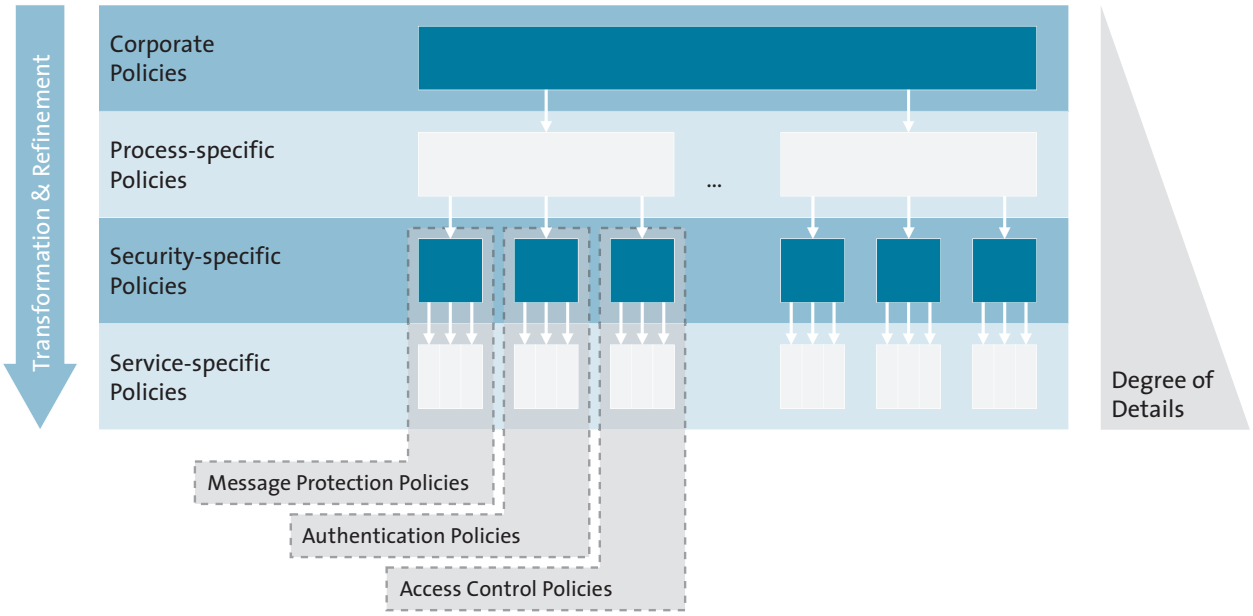


Abbildung 42: Policy Transformation und Refinement

Policy Administration

Die Policy Administration setzt innerhalb der Policy-Hierarchie in der Ebene auf, ab der eine automatische Transformation bis zu den IT-interpretierbaren Richtlinien gegeben ist. Da durch die Veränderung von Richtlinien die Möglichkeit einer direkten Beeinflussung der IT-Systeme besteht, müssen die Administrationsaktivitäten mit gesonderten Maßnahmen vor unerlaubter Benutzung geschützt werden. Insbesondere ist festzulegen und durchzusetzen, wer welche Richtlinien ändern darf. Die Verteilung der Administrationsaufgaben auf verschiedene Personen erscheint hier sinnvoll. Zudem sollten über geeignete Auditing-Mittel alle Aktivitäten protokolliert werden. Es ist ratsam, neue bzw. geänderte Richtlinienbestände nur durch gesonderte Freigabeprozesse mit qualitätssichernden Maßnahmen freizugeben, da die Richtlinien einen Einfluss auf die Verfügbarkeit der IT-Systeme haben werden. Zudem kann eine Versionierung von Richtlinien die Wiederherstellung im Fehlerfall unterstützen.

Top-down vs. Bottom-up

Der Top-down-Ansatz ist die bevorzugte Art und Weise Richtlinien zu entwickeln, um diese entsprechend den

geschäftlichen Anforderungen auszurichten. Um aber die Fähigkeiten der jeweiligen IT-Systeme in Bezug auf die Sicherheitsverfahren berücksichtigen zu können, wird man um eine Bottom-up-Betrachtung nicht herumkommen. Schließlich gilt auch für die Policy Definition der Grundsatz, dass sich die Richtlinien an den Fähigkeiten der IT-Systeme orientieren müssen. Dabei sollte die Realisierung mit vertretbarem Aufwand erfolgen. Entsprechend ist eine Lösung von Konflikten zwischen den geschäftlichen Anforderungen und der möglichen Umsetzung in der IT ein wichtiger Bestandteil des Definitionsprozesses. Zudem berücksichtigt erst die Bottom-up-orientierte Entwicklung von Richtlinien deren mögliche Wiederverwendung. Wird innerhalb einer SOA beispielsweise ein Service mehrfach, aber in verschiedenen Sicherheitskontexten (z. B. Internet oder Intranet) verwendet, sind die folgenden Lösungsstrategien anwendbar:

Der Service ist in Abhängigkeit vom aufrufenden Konsumenten in der Lage, verschiedene Sicherheitseinstellungen zu verwenden. In diesem Fall können die Richtlinien unabhängig voneinander entwickelt werden. Die Auswahl der anzuwendenden Richtlinien ist zur Laufzeit vom aktuellen Kontext abhängig.

Der Service kann aufgrund von technischen Restriktionen nur einen Satz von Sicherheitseinstellungen verwenden. Entweder werden die stärksten Richtlinienanforderungen verwendet oder es müssen entsprechende Kompromisslösungen gefunden werden. In jedem Fall ist dies aber nur mit einer Bottom-up-Betrachtung realisierbar.

Die Diskussion Top-down vs. Bottom-up in der Entwicklung von Richtlinien zeigt Parallelen zum Thema Entwicklung von Webservices. Hier stehen die Services mit geschäftlicher Orientierung, die nach dem Top-down-Ansatz entwickelt wurden, den Services mit applikationsspezifischer Sicht gegenüber, die nach dem Bottom-up-Ansatz entwickelt wurden. Als ein akzeptabeler Weg hat sich auch in dieser Hinsicht die Kombination der beiden Ansätze erwiesen.

Policy Enforcement

Das Policy Enforcement ist für die Durchsetzung der Richtlinien auf der Service-Seite verantwortlich. Das bedeutet, dass die Request-Nachrichten oder die eigentlichen Aktionen abgewiesen werden, wenn diese nicht den Regeln entsprechen. Dazu benötigt der PEP die vollständige Nachricht oder mindestens Informationen über die Identifikation des Nachrichtensenders sowie die Aktions- und Ressource-Kennung. Diese Daten bilden die Grundlage für die Auswahl der relevanten Richtlinien sowie deren Analyse bezüglich ihrer Auswirkung. Dazu ist grundsätzlich der Zugriff auf ein Policy Repository notwendig. Die Auswertung der Richtlinien kann hinsichtlich des PEP auf zwei verschiedenen Varianten beruhen:

Der PEP hat einen direkten Zugriff auf die gespeicherten Richtlinien. Die Auswertung dieser Richtlinien bzgl. Relevanz und deren Auswirkung erfolgt innerhalb des PEP. Diese Variante eignet sich für Richtlinien, deren Informationen direkt für die Durchsetzung verwendet werden können, ohne dass sie eine komplexe Analysemethodik erfordern. Ein Beispiel hierfür sind Richtlinien für den Nachrichtenschutz. Diese geben Anforderungen (Algorithmus, Reihenfolge, Schlüsseltyp, ...) bzgl. der Verschlüsselung und Signierung vor, die dann direkt mit der Nachricht verglichen werden. Entspricht die Nachricht diesen Anforderungen, wird sie weitergeleitet. Anderenfalls wird die Nachricht abgelehnt.

Alternativ kann die Entscheidungsfindung, ob die Request-Nachricht bzw. die auszuführende Aktion aufgrund der vorgegebenen Richtlinien bewilligt oder abgelehnt werden muss, in eine externe Komponente/einen Service (PDP) ausgelagert werden. Bei dieser Variante übergibt der PEP spezielle Informationen an den PDP. Dies entspricht dem Architekturprinzip „Security as a Service“ und bietet die Möglichkeit der Entkopplung der Policy-Validierung von dem eigentlichen Service. Damit erfordert nur der PDP einen Zugriff auf die Richtlinien. Allerdings liefert der PDP als Ergebnis der Entscheidungsfindung nur ein „Urteil“. Für die Ausführung des Urteils ist dagegen der PEP verantwortlich.

In einigen Anwendungsfällen ist ein Policy Enforcement auch auf der Client-Seite zu realisieren. Beispielsweise könnte es aus Sicherheitsgründen erforderlich sein, dass der Client die empfangenen Response-Nachrichten in Bezug auf die Einhaltung der vereinbarten Richtlinien überprüft.

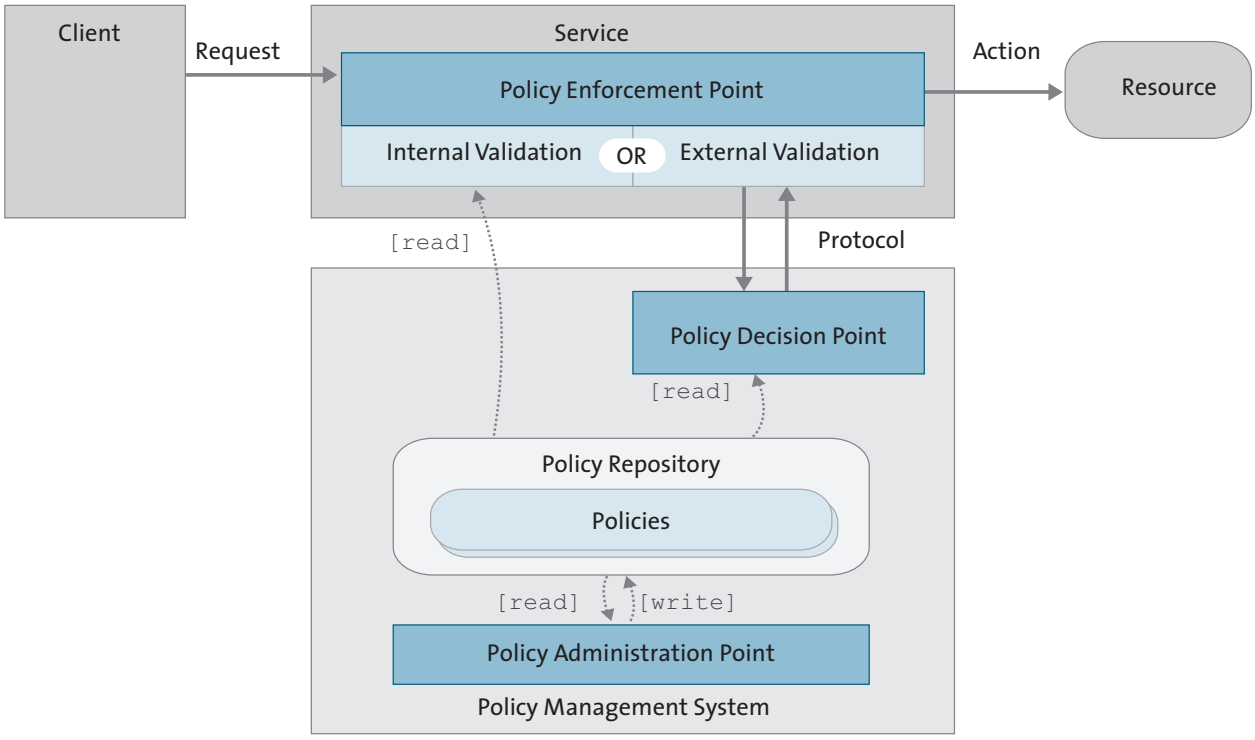


Abbildung 43: Policy Enforcement

7.3.4 Client Policy Provisioning

Innerhalb eines Client-Service-Szenarios ist es wichtig, die Service-Seite mithilfe einer Policy-Enforcement-Strategie gegen Sicherheitsverstöße zu schützen. Zusätzlich muss aber dafür gesorgt werden, dass die Clients als Nutzer der Services die service-spezifischen Richtlinien kennen, um sich regelkonform verhalten zu können. Basierend auf dem Prinzip der losen Kopplung gibt es im Webservice-Umfeld den WSDL-Standard, der den Clients diverse Service-Beschreibungen in einer allgemeinen Form zur Verfügung stellt. Darauf aufbauend besteht die Möglichkeit, auch Policy-Informationen mittels des WSDL zu den Clients zu transportieren. Im Fall von Richtlinien zum Nachrichtenschutz sowie für das Authentifizierungsschema übernimmt der Standard WS-Security Policy diese Aufgabe. Bei den WDSL-basierten Verfahren ist zu beachten, dass die Anforderungsabstimmung zwischen Client und Service nicht permanent erfolgt, sondern in der Regel vom Client nur in der Entwicklungs- bzw. Konfigurationsphase initiiert wird. Dies bedeutet aber, dass bei Policy-Änderungen auf der Service-Seite der Service-Aufruf des Clients scheitern wird, was dann wiederum zu Störungen im betrieblichen Ablauf führt. Es ist also notwendig, dass Policy-Management-Systeme nicht nur die Service-seitigen Richtlinien verwalten, sondern auch eine Lösung im Rahmen des Client Policy Provisioning anbieten. Stehen dem Client die gültigen Richtlinien zur Verfügung, kann der Policy Execution Point (PXP) die Umsetzung der Richtlinien übernehmen.

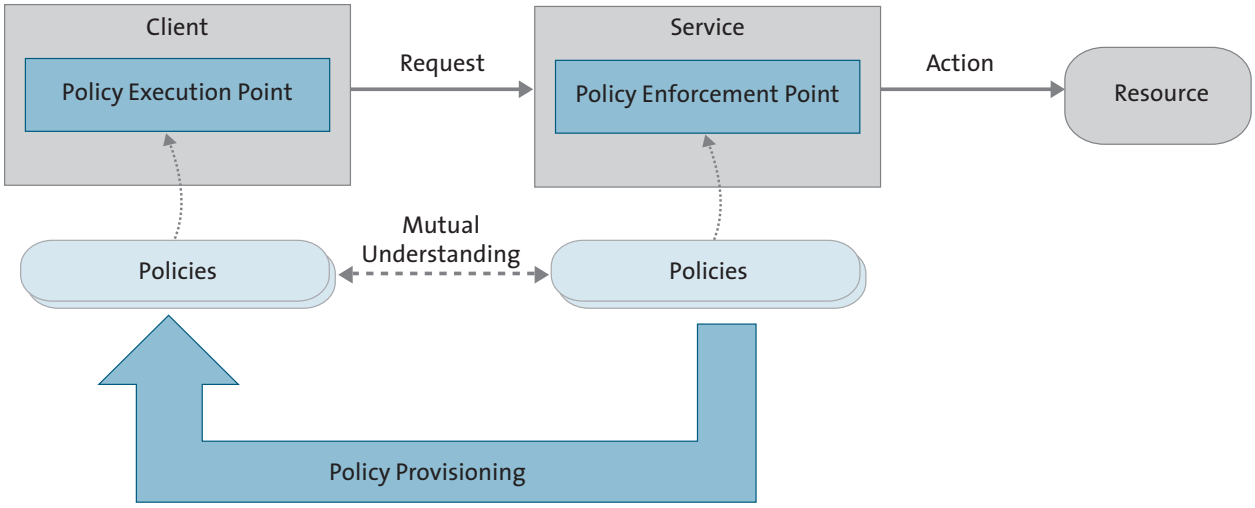


Abbildung 44: Client Policy Provisioning

7.3.5 Policy Deployment

Bei der Verteilung der Richtlinien in Bezug auf die PEPs können verschiedene Varianten unterschieden werden, die wiederum die Möglichkeiten der Administration beeinflussen. Ob die PEPs auf die Richtlinien direkt zugreifen oder deren Überprüfung an einen PDP ausgelagert wurde, spielt bei dieser Betrachtung keine Rolle. Folgende Varianten können unterschieden werden:

Service-spezifische Policies befinden sich lokal auf derselben Maschine bzw. auf demselben Container wie der Service. Die Administration ist für jeden Service (PEP) lokal und separat durchzuführen. Es ist allerdings ein erheblicher Aufwand notwendig, um beispielsweise einen Business-Prozess, der sich über eine Vielzahl von Services spannt, auf eine einheitliche Zugriffskontrolle einzustellen. Diese Variante kann als Ausgangslage betrachtet werden, bevor ein zentrales Policy-Management-System zum Einsatz kommt.

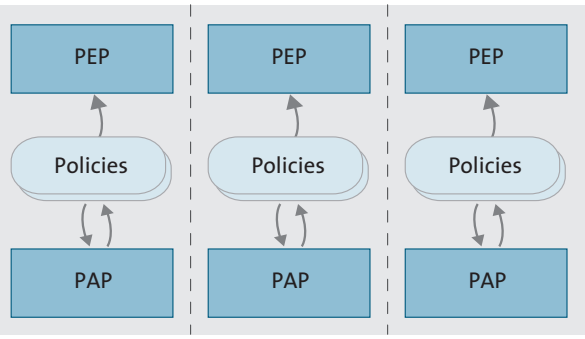


Abbildung 45: Variante 1 – lokale Policies, separate Administration

In der zweiten Variante liegen die Policies ebenfalls lokal vor, aber es ist eine zentrale Administration der verschiedenen Policy Repositories möglich (Single Point of Administration). Die Administration hat u. a. die Aufgabe, die Service-übergreifenden Richtlinien in allen Repositories zu synchronisieren.

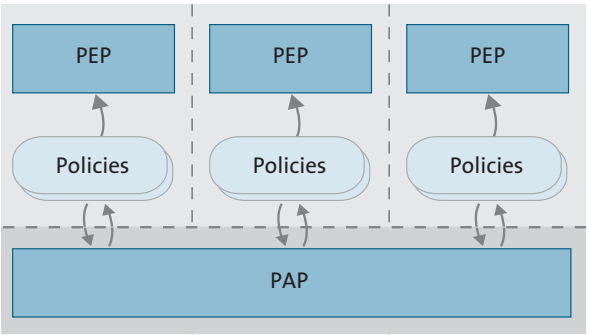


Abbildung 46: Variante 2 – lokale Policies, zentrale Administration

Die dritte Variante bietet die Möglichkeit, alle Richtlinien in ein gemeinsames und zentrales Repository zu speichern. Dadurch ist es ohne großen Aufwand möglich, serviceübergreifende Einstellungen zu verwalten. Nachteilig ist aber, dass die PEPs für jeden Enforcement-Vorgang einen Netzwerkzugriff benötigen. Abgesehen davon, dass Performance-Probleme entstehen können, ist dadurch auch kein Offline-Modus möglich.

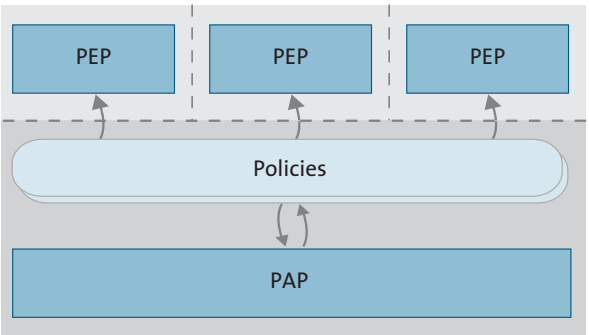


Abbildung 47: Variante 3 – zentrales Repository ohne Offline-Modus

Die vierte Variante hat ebenfalls ein gemeinsames und zentrales Policy Repository, auf dem die Administration aufsetzt. Dagegen benutzen die PEPs der jeweiligen Services nur eine lokale Kopie und sind somit nicht ständig vom Netzwerk abhängig. Damit ist ein Offline-Modus realisierbar. Die Verteilung der Richtlinien kann dabei nach dem Push-Prinzip oder nach dem Pull-Prinzip erfolgen. Beim Push-Prinzip werden geänderte Richtlinien automatisch von einer zentralen Instanz auf die lokalen Repositories verteilt. Dagegen kontrollieren die PEPs beim Pull-Prinzip selbstständig den aktuellen Stand ihrer Daten und übernehmen bei Bedarf deren Anpassung. Um das unerlaubte Ändern von Richtlinien in den lokalen Kopien zu verhindern, erfordern lokale Repositories jedoch zusätzliche Sicherheitsmaßnahmen (Last Mile Security).

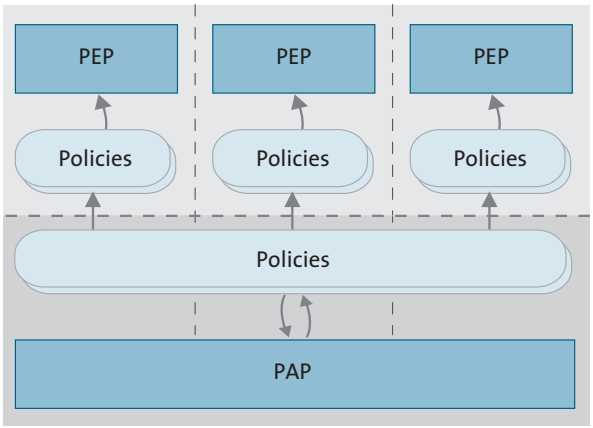


Abbildung 48: Variante 4 – zentrales Repository mit Offline-Modus

7.4 Risikomanagement in einer SOA-Welt

Der Einsatz moderner Technologien wie SOA führt immer wieder zu kontroversen und detailbelasteten Diskussionen. Trotz aller Versuche, Risiken möglichst objektiv darzustellen, entbrennen immer wieder regelrechte Glaubenskriege zwischen der Fach- und der IT-Seite. Nicht selten führen diese zu einer Blockade von neuen Technologien beziehungsweise stehen ihrer geordneten Einführung im Wege, ohne sie jedoch verhindern zu können. Kristallisationspunkt des Streites sind regelmäßig die empfohlenen organisatorischen und technischen Maßnahmen sowie ein fehlendes gemeinsames Verständnis der Ziele. Sehr häufig fehlt der übergeordnete strategische und organisatorische Rahmen, um für die fachlichen Aspekte der Geschäftsseite und die fachlichen Aspekte der IT-Seite gleichermaßen belastbare Entscheidungen zu erhalten.

Gerade im Kontext SOA wird dies verstärkt deutlich. SOA ist geeignet, Geschäftsprozesse in einem bisher schwer erreichbaren Ausmaß mit der IT verwachsen zu lassen. Ein gemeinsames Verständnis der Geschäftsprozesse ist einfacher erreichbar als in bisherigen IT-Lösungsansätzen, so ist ein effizienterer Einsatz der IT möglich. Die IT wandelt sich von einem reinen Stützprozess zum integrierten Bestandteil des Geschäftsprozesses. Gegenseitige Abhängigkeiten nehmen stark zu. Um eine effizientere Nutzung der IT im Rahmen von "Business Alignment" zu erreichen, ist dies sogar explizites Ziel von SOA. Als zwingende Folge ist auch das klassische Risikomanagement für die Geschäftsprozesse mit dem IT-Sicherheitsmanagement enger zu verzahnen. Bleibt dies aus, sind Missverständnisse und unnötige Reibereien zwischen den Trägern des fachlichen Geschäfts und der Informationstechnologie unvermeidbar. Schlimmer noch, tatsächlich existierende Risiken werden falsch oder gar nicht bewertet, potenzielle Chancen werden dagegen verpasst.

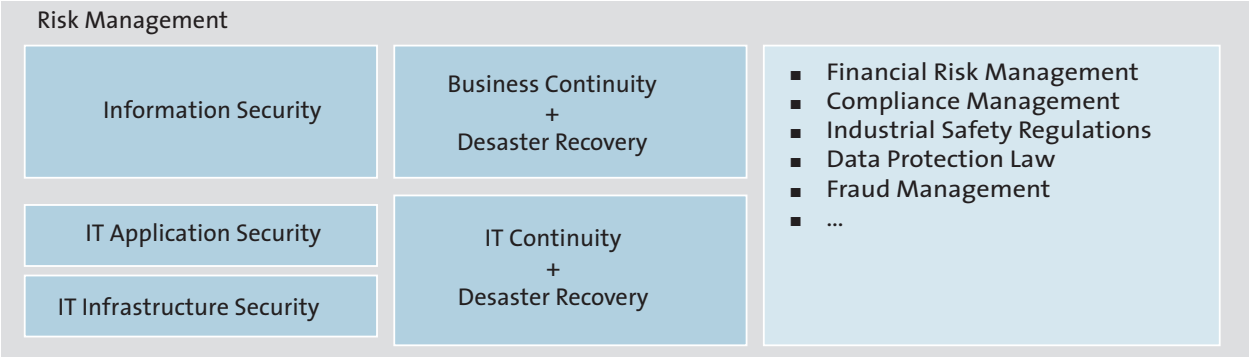


Abbildung 49: Risk Management

Schutzbedarf

Um die aus SOA erwachsenden Risiken bewerten zu können, muss allen Beteiligten bekannt sein, vor welchen „Gefahren" man sich eigentlich schützen will. Diese scheinbar einfache Frage nach dem Schutzbedarf ist eine der am schwierigsten zu beantwortenden Fragen überhaupt. Der Schutzbedarf ergibt sich letzten Endes aus den einem Prozess zugrunde liegenden Informationen. Hier bedarf es einer intensiven Abstimmung von Fach- und IT-Seite unter Koordination des Risikomanagements.

Bedrohungen

Es ist wichtig zu verstehen, dass SOA nicht eine fixierte Technologie ist, sondern vielmehr eine natürliche, evolutionäre Sichtweise auf Software-Architektur „verkörpert". Dementsprechend liegen die neuen Bedrohungen nicht in völlig neuen technologischen Herausforderungen, sondern darin, in einer veränderten Kultur Software zu erstellen und diese Software in diese neue Kultur zu integrieren. Diese neue Kultur muss in den beteiligten Köpfen umgesetzt werden. Um es zu wiederholen: Neu ist an dieser Stelle die geforderte stärkere Einbeziehung der geschäftsfachlichen Seite. Die Bedrohungen durch neue Protokolle und Technologien bedürfen zwar ebenfalls einer gründlichen Würdigung, sie sind aber in großen Bereichen durch seit Jahren verfügbare Checklisten behandelbar.

Risiken

Risiken ergeben sich aus den möglichen Bedrohungen, dem Schadensmaß und dessen Eintrittswahrscheinlichkeit. Genauso wenig wie das dogmatische Ausblenden der "nur" stützenden IT im fachlichen Risikomanagement des Unternehmens die Lösung ist, ist dass Aufzählen von technologischen und organisatorischen

Verwundbarkeiten seitens des IT-Security-Managements die richtige Antwort auf Bedrohungen. Von dem Umgang mit Risiken handelt das Risikomanagement. Dieses muss die entstehenden Risiken ganzheitlich und interdisziplinär bewerten und Vorgaben machen, die auch für die IT gelten und umsetzbar sind.

Für die an die Geschäftsprozesse anwachsende Informationstechnologie bedeutet dies eine Ausweitung der einzubeziehenden Schadensdimensionen bei ihren Sicherheitsanalysen. Die zu betrachtenden Risiken werden vielfältiger in ihrer Art, teilweise sind sie losgelöst von der zugrunde liegenden Technologie und Organisation.

Ein einfaches Formulieren der Sicherheitsanforderungen in Form eines Service-Levels für die Verfügbarkeit durch die Geschäftsseite reicht häufig nicht aus.

Wie mit diesen Risiken umgegangen wird, hängt maßgeblich davon ab, ob die tatsächlich relevanten Schadensaspekte erkannt werden und ihre Auswirkungen in Form der möglichen Verlusthöhe und der zu erwartenden Verlusthäufigkeit richtig eingeschätzt werden. Hierzu müssen Fach- und IT-Experten interdisziplinär gemeinsame Ergebnisse finden.

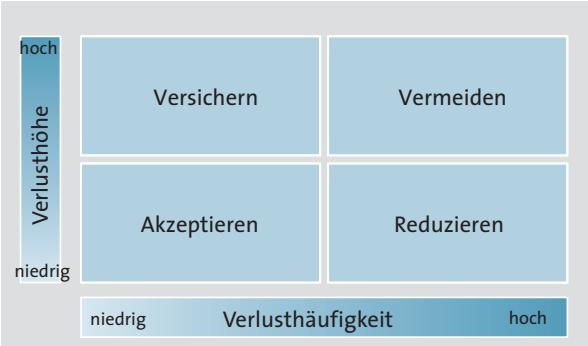


Abbildung 50: Handlungsmöglichkeiten zum Umgang mit Risiken

Mögliche zu berücksichtigende Schäden lassen sich dabei grob in vier Gruppen fassen:

- direkter monetärer Schaden (z. B. Systemausfälle, Umsatzeinbußen, Vertragsstrafen)
- indirekter monetärer Schaden (z. B. Verlust an Kunden, Rufschädigung, Schädigung der Allgemeinheit, Schadensersatzklagen, notwendige PR-Maßnahmen, kostenintensive nachträgliche Korrektur)
- Verstoß gegen gesetzliche und regulative Auflagen (z. B. strafrechtliche Verfolgung oder Ausschluss aus Märkten, Kosten für die juristische Aufbereitung von Verstößen)
- Schäden an Personen (z. B. durch fehlerhafte Produkte oder Arbeitsabläufe)

Die zu berücksichtigenden Risiken werden in der klassischen Organisation des Risikomanagements häufig von spezialisierten Einheiten mit wenig Kontakt untereinander betrachtet. Das Gewicht der Informationstechnologie und damit die Abhängigkeit der Geschäftsprozesse von der IT wachsen jedoch rasant; die daraus bedingte enge Verzahnung von Fach- und IT-Seite ist häufig noch nicht ideal bzw. scheitert vielfach am Fehlen einer gemeinsamen Sprache bzw. Metrik zur Bewertung von Schäden und Risiken.

Das IT Continuity Management zeigt einen brauchbaren Weg, den Prozessgedanken im Risikomanagement der IT zu stärken. Wichtig ist hierbei, tatsächlich den zu schützenden Prozess zu betrachten und nicht ausschließlich in technologischen Lösungen zur Erhöhung der Redundanz wie zum Beispiel Raid-Systemen oder Server-Clustern zu denken.

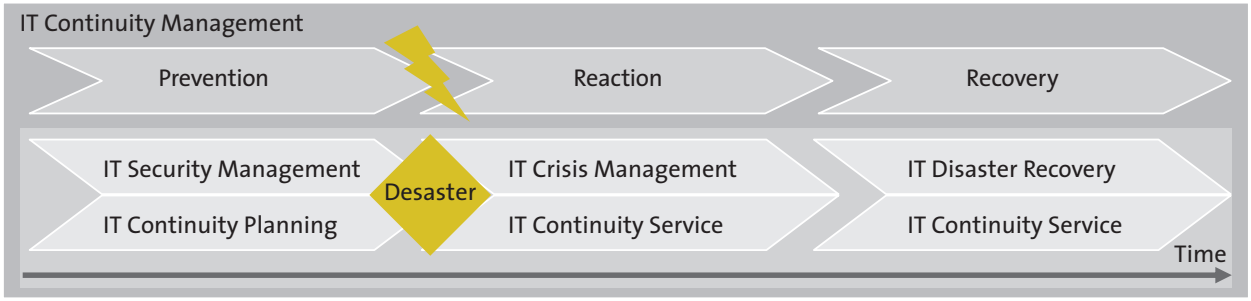


Abbildung 51: IT Continuity Management

Ideen für ein modernes Risikomanagement in einer SOA-Welt:

- Eine moderne Sicherheitsorganisation orientiert sich an den Geschäftsprozessen der Unternehmung, der sie dienen soll. Ändern sich Prozesse, so muss die Organisation diese Veränderungen angemessen reflektieren. Die Kommunikation zwischen den beteiligten Sicherheitsorganisationen der Fach- und IT-Seite muss gefordert und gefördert werden.
- SOA verlangt die Betonung des Prozessgedankens in der IT-Sicherheit. Das Denken in klassischen Kategorien der IT-Infrastruktur und IT-Organisation deckt nur noch einen Teil der Anforderungen ab. SOA basiert auf Prinzipien, die zu einer transparenteren und standardisierten Sicherheitslage führen. Die Umsetzung gelingt am besten durch den Ansatz „Sicherheit als ein Service“ innerhalb einer SOA. Der Service-Gedanken von SOA sollte daher konsequent auf Ihre IT-Sicherheitsorganisation überragen werden.
- Alle Formen von möglichen Schäden müssen in einer Sicherheitsanalyse berücksichtigt werden. Alle Disziplinen einer Organisation sollten am Sicherheitsteam beteiligt werden. Hieraus ergibt sich ein stark wachsender Bedarf der Kommunikationsfähigkeit auf der Fach- und IT-Seite.
- Bei Bedarf sollte auch auf Expertise von extern zurückgegriffen werden. Die zwangsläufig „subjektiven“ Ergebnisse eines externen Sicherheitsberaters müssen durch das eigene Risikomanagement gewichtet werden. Diese für ein Unternehmen spezifische Gewichtung muss auf einer schriftlich fixierten Strategie, Vorgehensweise und zugehörigen Metriken basieren. Fehlt dieser Rahmen, werden Strategie und Vorgehensweise zuerst erstellt!
- Eine Prozessorientierung in der Sicherheit lässt sich vergleichsweise einfach über das Thema Business Continuity und Desaster Recovery angehen. Welche Verbindung gibt es zwischen Business Continuity, Information Security und Risikomanagement in einer Organisation? Woran merkt man, dass eine ausreichende Abstimmung erfolgt?
- Technisch/organisatorische Richtlinien sind sicher ein Muss, sie sollten aber nicht das alleinige Verständnis einer IT-Sicherheitsorganisation ausmachen. Für einen ersten schnellen Check der Ausrichtung einer IT-Sicherheitsorganisation könnte man beispielsweise deren wesentliche Publikationen betrachten. Findet sich hier neben technisch/organisatorischen Checklisten auch der Gedanke eines Security Service für das Geschäft einer Unternehmung? Gibt es zielgruppenbezogene Dokumente für die Fachseite? Wie wird der Service-Gedanke ausgelegt? Wird er gelebt?
- Nach welchen Kriterien, Metriken und nach welcher Methodik werden Maßnahmen zum Umgang mit Risiken innerhalb der IT-Sicherheitsorganisation bewertet. Um dies herauszufinden, sollte mit allen “Teilen” einer Sicherheits-Organisation geredet werden. Werden zumindest vergleichbare Kriterien, Metriken sowie ein vergleichbare Methodik eingesetzt? Kann man die Arbeitsgrundlage vereinheitlichen?
- Versteht sich eine Sicherheitsorganisation eher als Revision, die Vorgaben macht und gegen diese prüft, oder ist sie service-orientiert, eher beratend und unterstützend unterwegs?

7.5 Security-Implementierungen

7.5.1 Architekturpattern

Lösungen, die heute auf Basis einer SOA implementiert werden, sind auch immer verteilte Systeme. Zurzeit werden diese Lösungen meist noch ausschließlich unternehmensintern genutzt, aber es ist abzusehen, dass diese Beschränkung in nicht allzu ferner Zukunft fallen wird. Services werden also zukünftig über Unternehmensgrenzen hinweg verwendet werden. Somit sind die auch bisher geltenden Basisanforderungen wie Authentifizierung, Autorisierung, Integrität, Vertraulichkeit und Nichtabstreitbarkeit auf verteilte Anwendungen im SOA-Umfeld anzuwenden.

Zur Umsetzung dieser Sicherheitsaspekte gibt es eine Reihe von akzeptierten Security-Standards aus dem Umfeld einiger etablierter Standardisierungsgremien. Diese Standards können schon bei der Implementierung der Services genutzt werden, sind aber relativ komplex in der Anwendung und haben zu viele Freiheitsgrade bei der Umsetzung.

Aus diesem Grund haben sich neben der Implementierung der sicherheitsrelevanten Funktionen direkt in der Applikation zwei Ansätze herausgebildet, die auf diesen Security-Standards basierend die Implementierung einer sicheren SOA ermöglichen wollen:

- Security as Infrastructure versucht dabei, die sicherheitsrelevanten Dienste als Teil der Infrastruktur zu implementieren, häufig als sogenannte „Appliance“, die an zentraler Stelle zum Einsatz kommen muss.
- Security as a Service greift den SOA-Gedanken auf und stellt die Sicherheitsfunktionen zentral als Service bereit. Diese können dann von allen anderen Komponenten einer SOA genutzt werden.
- Mit Embedded Security bezeichnet man in diesem Umfeld den Ansatz, die Security-Funktionen durch die Applikation selbst zu implementieren. Dies kann sowohl bei verteilten als auch monolithischen Implementierungen erfolgen. Dieser Ansatz kann selbstverständlich auch in einer SOA umgesetzt werden, dies ist aber nicht empfehlenswert.

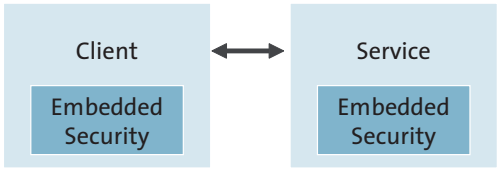
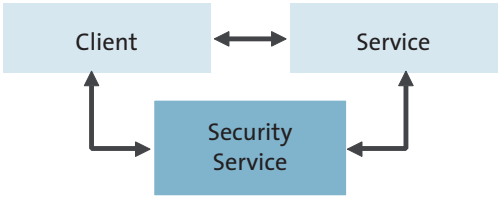
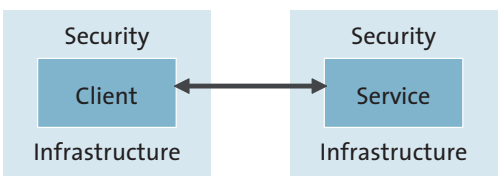
	Realisierung der Sicherheit innerhalb der Business Service	+ Unabhängig von Infrastruktur - Keine zentrale Administration - Jeweils proprietare Lösung der Security (Kompatibität) - Keine Trennung von Applicationen und Security
	Zentrale Bereitstellung von Funktionen zur Sicherheit	+ Wiederverwendbarkeit der Security + Zentrale Management der Security + Applikationslogik und Security sind getrennt - Vertrauen zum Security Service muss vorhanden sein
	Sicherheit liegt komplett in der Infrastruktur	+ Applikation und Security sind getrennt + Security Implementierungist individuell bekannt - Security Implementierungist nur individuell bekannt - Individuelle Infrastruktur gefährdet die Kompatibilität

Abbildung 52: Entwurfsmuster für SOA Security

„Security as a Service“ bietet dabei eine nahtlose Integration des SOA-Gedankens und wird im Folgenden als bevorzugte Lösung beschrieben. Dies setzt allerdings voraus, dass die zugrundeliegende Infrastruktur bereits den Sicherheitsanforderungen entspricht. Lücken an dieser Stelle haben daher Lücken in den darauf aufsetzenden Diensten zur Folge. Ausserdem ist zu berücksichtigen, dass die Verfügbarkeit und die Performance dieser Services höchsten Anforderungen genügen muss. Performance Probleme bei den Security Services beeinträchtigen alle Applikationen, die diese einbinden.

7.5.2 Security Services für eine SOA

Bei der Definition der notwendigen Dienstgruppen kann auf Erfahrungen mit schon im Einsatz befindlichen verteilten Systemen zurückgegriffen werden. Im Folgenden sind nur die Security-relevanten Dienstgruppen aufgeführt. Hier hat sich eine Aufteilung in folgende Bereiche als sinnvoll erwiesen:

Security Services: In den Security Services befinden sich alle grundlegenden Funktionen, hier werden die Crypto-Algorithmen und Basisprotokolle angeboten. Außerdem werden hier an zentraler Stelle die im System nutzbaren Authentifizierungsprotokolle zur Verfügung gestellt. Die Bandbreite reicht dabei von User-ID/-Passwort über die Unterstützung von Smartcards bis zu biometrischen Verfahren. Die Art und Qualität des Authentifizierungsverfahrens wird dabei meist in der dann etablierten Session des Benutzers abgelegt. Diese Information steht danach für die Autorisierung zur Verfügung.

Auch die Autorisierungsverfahren werden auf dieser Ebene implementiert. Diese haben einen Sonderstatus im System, da sie sowohl zur personenbezogenen Autorisierung genutzt werden als auch für die Autorisierung der einzelnen Services untereinander (keine Benutzerinteraktion). Die Autorisierung wird meist als Policy Enforcement bezeichnet, da die Entscheidungen häufig auf Policies basieren, in denen die Security-Anforderungen des Service hinterlegt sind. Diese werden gegen die Benutzerinformationen und eventuell weitere (Meta-)Daten wie z. B. die Qualität der Authentifizierung geprüft. Erst dann wird entschieden, ob der Zugriff gestattet wird.

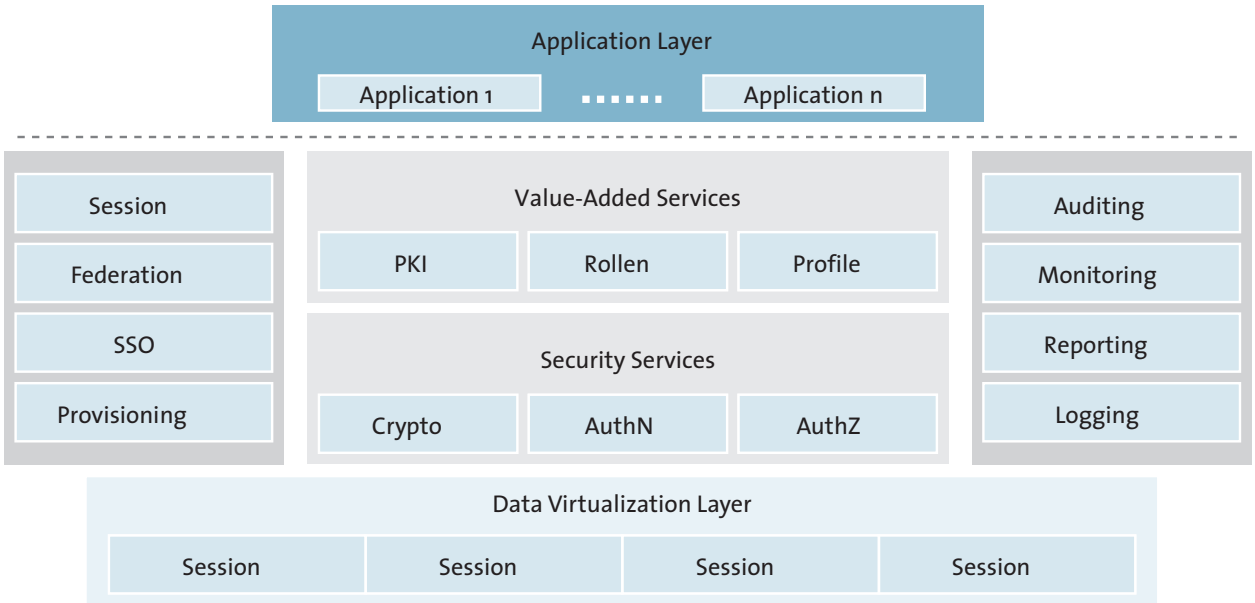


Abbildung 53: Beispielimplementierung einer SOA

Value-added Services stellen Dienste wie Benutzer, Rollen und Profile Management zur Verfügung. Wenn notwendig, ist hier die Public-Key-Infrastruktur (PKI) angesiedelt.

Das Benutzermanagement ist kein originärer Security Service, es ist aber für den sicheren und effizienten Betrieb unabdingbar. Das Benutzermanagement kümmert sich um die workflowgestützte automatisierte Provisionierung der Benutzer auf alle nachgelagerten Systeme. Dies bezieht die attributgestützte Rechtevergabe in Form von Zuweisung von Policies und/oder Rollen mit ein. Neben dem zeitnahen und vollständigen Anlegen der Benutzer auf allen vorgesehen Systemen ist das zeitnahe Entziehen (De-Provision) des Benutzers fast noch wichtiger. Die hierzu notwendige Workflow Engine sollte als Basisdienst der SOA vorhanden sein, dieser wird von den Security Services genutzt.

Mittels des Rollenmanagements werden die in den Systemen vorhandenen Rollen erfasst, vereinheitlicht und in ein globales Rollenmodell gewandelt. Dabei muss es möglich sein, sowohl ein flaches Rollenmodell für einfache Implementierungen als auch komplexere mehrdimensionale Modelle abbilden zu können. Werden im Rollenmanagement auch Systemrollen gehalten, muss hier eine Abbildung auf das jeweilige Zielsystem definiert werden können.

Die definierten Rollen können nun mittels des Benutzermanagements für die Zuweisung und – sehr viel wichtiger – für den Entzug von Rechten genutzt werden.

Das Profile Management verwaltet die Sicherheitsanforderungen der einzelnen Services und Applikationen mit Bezug auf unterschiedliche Kontexte. In einem Profil sind alle für einen zu implementierenden Service notwendigen Subservices und ihre Kommunikationsbeziehungen verzeichnet. Damit können ihnen, basierend auf dem durch die Risikoanalyse ermittelten Bedrohungspotenzial, Sicherheitsmechanismen unterschiedlicher Güte und/oder spezielle Verfahrensweisen zugewiesen werden.

Zur Verschlüsselung oder Signierung von Nachrichten werden Zertifikate benötigt. Diese können von einer eigenen PKI erstellt werden oder von einem externen Dienstleister bezogen werden. Es ist darauf zu achten, dass die PKI eine Schnittstelle zur Integration in Webservices-Protokolle hat.

Session Services: Der Session Service beschäftigt sich mit der Generierung und dem Verwalten der User Sessions, dem Single Sign-on und dem Single Log-out. Der Session Service ist kein originärer Security Service, da aber im Besonderen der Punkt Single Log-out Auswirkungen auf die Security des Gesamtsystems hat, soll er hier erwähnt werden.
Muss man über Unternehmensgrenzen hinaus tätig werden und akzeptiert die Forderung nach durchgehender Authentifizierung und Autorisierung, werden die Federation Services benötigt. Prinzipiell geht es darum, autonomen Organisationen ohne zentrale Instanz die Zusammenarbeit zu ermöglichen, indem ein Mechanismus gefunden wird, mit dem existierende Benutzerkonten mit Zustimmung des Benutzers in Relation gesetzt werden können. Der Federation Service arbeitet eng mit dem Session Service zusammen, um auch organisationsübergreifend Sessions aufbauen zu können.

Logging Services: Jede Aktion innerhalb des Systems muss protokolliert werden, dies schließt Aufrufparameter und Rückgabewerte ein.

Der Logging Service ist kein originärer Security Service. Da jedoch häufig kein einheitliches Logging-Verfahren existiert und auch die Formate sich unterscheiden, wird hier die Forderung nach einem einheitlichen, übergreifenden und revisionssicher implementierten Logging Service aufgestellt.

Monitoring Services: Dieser Service dient der Überwachung aller Aufrufe, dem Tracing im Fehlerfall sowie der Überwachung der Services. Er ist als Service implementiert, um ihn auch auf dem Service-Level (ohne Benutzerinteraktion) im Zugriff zu haben.

Audit Services dienen der Speicherung aller relevanten Informationen wie Service-Aufrufe, Ergebnisse etc. Die

gesamte Historie ist abgelegt und kann im Rahmen interner oder externer Prüfungen abgerufen werden.

Reporting Services: Basierend auf den Logging-Daten müssen (nahezu) beliebige Reports möglich sein. Dieser Service ist damit die Grundlage für die Entwicklung eines Management-Cockpits für die Sicherheitslage. Der Zugriff darf dabei natürlich nur berechtigten Benutzern gestattet sein. Dieser Dienst ist als Service ausgelegt, um ihn anderen Services zur Verfügung stellen zu können. Hier werden Benachrichtigungsmechanismen wie E-Mail oder SMS implementiert.

Häufig sind weite Teile der oben angesprochenen Services durch Identity- und Access-Management-Systeme implementiert. Das Identity Management System kümmert sich dabei um die Bereitstellung der Benutzer und aller relevanten Informationen. Das Access Management System kümmert sich um die Bereiche Authentifizierung und Autorisierung. Idealerweise stützen sich dabei beide auf dieselben Basisdienste und passen sich auch in das Monitoring-, Audit- und Reporting-Framework ein.

Die eigentliche Herausforderung einer solchen Architektur besteht in der Einbindung der bestehenden Applikationen und Standardprodukte in neue SOA-basierte Dienste. Diese Kombination erschwert die Umsetzung von Sicherheitsaspekten auf Ebene der einzelnen Dienste. Als Ausweg bietet sich hier die Kapselung der Altanwendungen mit anschließender Integration in das Gesamtkonzept an. „Security as a Service“ stellt dabei einen vielversprechenden Lösungsansatz zur Umsetzung dar, mit dem die Anwendungen im SOA-Umfeld erstellt werden können. Sollten sich im Laufe der Entwicklung oder des Betriebes Neuerungen bei den Security-Komponenten ergeben, sind diese durch den modularen Aufbau sofort im Gesamtsystem präsent und können sofort eingebunden werden.

7.5.3 Lösungsansätze

In diesem Abschnitt werden mögliche Lösungsansätze aufgezeigt. Diese können dabei den verschiedenen Schichten Geschäftsprozess, Applikation und Middleware (die Ebene Netzwerk wird hier nicht eigens betrachtet) zugeordnet werden.

Geschäftsprozess

Schon auf der Geschäftsprozessebene sind die Anforderungen bezüglich Vertraulichkeit, Integrität und Verfügbarkeit zu formulieren. Wie bereits erwähnt müssen sich dazu die Fachbereiche und die IT-Abteilungen auf eine gemeinsame Sprache und Metrik verständigen. Einen wichtigen Messparameter stellt das Risiko dar. Um diesen Parameter ermitteln zu können, muss man z. B. im Risikomanagement verankerte Schadens- und Häufigkeitsklassen verwenden und einen Bedrohungskatalog entwickeln, der sowohl von den Fachbereichen als auch von den IT-Abteilungen verstanden und akzeptiert wird.

Applikation

Im Übergang zu einer SOA werden bestehende Applikationen häufig mit Webservice-Schnittstellen versehen, um in eine SOA-Umgebung eingebunden werden zu können. Diese Schnittstellen bieten andere Angriffsmöglichkeiten als im Umfeld herkömmlicher Applikationen; diese müssen analysiert werden. Typische Angriffe auf XML-Basis können durch den Einsatz von Web Application Firewalls, die auch als XML Gateways fungieren, verhindert werden. Ob ein XML Filtering als Security Service oder als Komponente der Infrastruktur realisiert wird, hängt von mehreren Parametern wie Kosten, Zeit für die Umsetzung, Anforderungen an die Wiederverwendbarkeit des XML-Filters oder die Verfügbarkeit ab.

Zudem ist zu überlegen, ob auf Applikationsebene eine Verschlüsselung oder Signatur notwendig ist. In Bezug auf Webservices kann die Verschlüsselung auf SOAP- bzw. XML-Ebene erfolgen. Dabei können auch nur Teile der Nachricht verschlüsselt oder signiert werden. Die Verschlüsselung oder Signatur kann sowohl von der Applikation selbst oder durch entsprechende Webservices vorgenommen werden.

Wenn die Umsetzung einer SOA schon etwas fortgeschritten ist, wird der Einsatz eines Enterprise Service Bus (ESB) meist unvermeidlich, um die implementierten Services verwalten zu können. Der ESB wird als Vermittlerschicht zwischen den Webservices eingesetzt. Dadurch entstehen eine ganze Reihe neuer Fragestellungen bezüglich der Sicherheit, die mithilfe der genannten Security Services gelöst werden können. Hier einige Beispiele:

- Bei asynchroner Kommunikation werden Queues oder Buffer als Zwischenspeicher für Messages verwendet. Diese können vertrauliche Daten beinhalten, die vor unautorisiertem Zugriff geschützt werden müssen. Aus diesem Grund kann es notwendig sein, Buffer des ESB zu verschlüsseln.
- Für Transaktionssicherheit muss der ESB zustandsorientiert arbeiten. Weiter müssen Transaktionen geeignet autorisiert werden. Hier benötigt man Authentisierung und rollenbasierte Vergabe von Berechtigungen – nicht nur für Personen, sondern auch für Webservices.
- Auf Service-Ebene ist die Kommunikation zu regeln, z. B. kann eine Kommunikation zwischen zwei Services nicht erlaubt sein. Die Kommunikationskanäle werden hier durch Regeln oder Policies des ESB konfiguriert.
- Webservices rufen über den ESB andere Webservices ab. Hier ist die Authentizität von Webservices, z. B. über kryptografische Methoden (auch auf XML-Basis), sicherzustellen.

Die genannten Lösungsansätze müssen zudem in die Unternehmenskultur passen und durch operative Prozesse (ITIL-Prozesse wie Patch-, Config-, Change- und Incident-Management) unterstützt werden.

7.6 Wechselwirkung zwischen Performanz, Verfügbarkeit und Sicherheit in einer SOA

Ziel

Die Aspekte Verfügbarkeit und Performanz in einer SOA-Umgebung stehen in Wechselwirkung mit den Sicherheitsfunktionen. Wir betrachten hier speziell die Auswirkungen auf die Komponenten, die Sicherheitsfunktionen abbilden. Wir verstehen hierbei unter einer Sicherheitsfunktion nicht zwangsweise die Implementierung von Sicherheit als „Security as a Service“.

Diese Anforderungen sind nicht grundsätzlich neu, weil sie im Kontext einer SOA stehen. Sie erfahren jedoch in einer SOA-Umgebung teilweise eine neue Bedeutung und Gewichtung. So werden in einer SOA zum Beispiel kryptografische Funktionalitäten tendenziell stärker benötigt (z. B. Verschlüsselung der durch die verteilte Architektur notwendigen Kommunikation) und das Security Logging bekommt einen deutlich höheren Stellenwert. Durch die verteilte Architektur haben Sicherheitsfunktionen wie Authentisierung oder Autorisierung stärkere Auswirkungen auf die Performanz und Verfügbarkeit, z. B. bei der Überprüfung von Zertifikatsketten und Vertrauensbeziehungen zwischen den SOA-Komponenten. Diese erhöhten Anforderungen gelten dabei sowohl für den Einsatz von SOA innerhalb eines Unternehmens als auch im kollaborativen Umfeld mit gegebenenfalls neuen Betriebsmodellen (öffentliche Service-Plattformen).

Dieses Kapitel betrachtet eine Reihe dieser Themenfelder und versucht die Leser, auf kritische Bereiche hinzuweisen und ihnen Lösungsvorschläge zu nennen.

Security-Komponenten und -Funktionen

In einer SOA erfüllen bestimmte „Security-Komponenten“ Security-Funktionen.

Security-Komponenten sind zum Beispiel

- bzgl. Policy: Policy-Enforcement-/-Decision-Punkte sowie Policy-Administration-Punkte (PEP, PDP, PAP)
- Authentisierungsdienste
- Ver- und Entschlüsselungskomponenten
- Logging
- PKI
- Identity-Management-Systeme

Beispiele für PEP und PDP sind:

- Reverse-Proxy zur Benutzerautorisierung, Verzeichnisdienst, Access Control Server, XML Gateways
- Zentraler Verzeichnisdienstserver, Managementkonsolen zur Zugriffsrechteverwaltung
- Enterprise Service Bus (speziell die Komponenten Registry, Repository und Content/Security-based Routing)

Anforderungen

An diese Komponenten und Operationen werden Anforderungen bezüglich verschiedener Themenfelder gestellt. Die Themenfelder liegen weitgehend im Bereich der sogenannten nicht funktionalen Anforderungen (non-functional requirements). Sie betreffen nicht die grundlegende Funktion der SOA-Security-Infrastruktur, wohl aber die Rahmenbedingungen, unter denen diese betrieben wird. Sie haben somit entscheidenden Einfluss darauf, ob die Infrastruktur unter Praxisbedingungen die an sie gestellten Anforderungen erfüllt.

Die Themen untergliedern sich im Folgenden in die Bereiche:

- Performanz
- Verfügbarkeit
- Security

Die Themenfelder im Einzelnen

Performanz in einer SOA - Übersicht

Die Wechselwirkungen zwischen Performanz und Sicherheit in einer SOA sind in zweierlei Hinsicht wichtig:

(1) Auswirkungen der SOA-Sicherheit auf die Performanz:
Die speziellen Eigenschaften einer SOA führen zu speziellen Sicherheitsanforderungen. Werden diese Anforderungen erfüllt, hat das Auswirkungen auf die Performanz. Diese Auswirkungen werden im Folgenden beschrieben, zusätzlich werden Lösungsansätze vorgestellt.

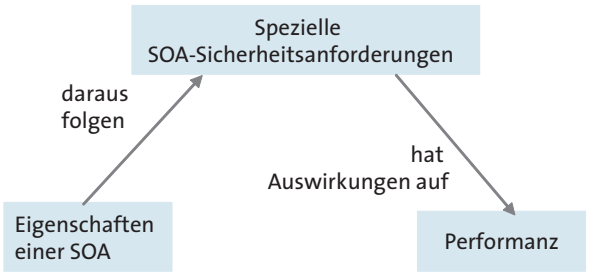


Abbildung 54: Auswirkungen der SOA-Sicherheit auf die Performanz

(2) Auswirkungen der Performanz auf die Sicherheit:
Aufgrund der Performanz-Anforderungen werden Entscheidungen über den Aufbau einer SOA getroffen. Dieser Aufbau hat wiederum Auswirkungen auf die Sicherheit.

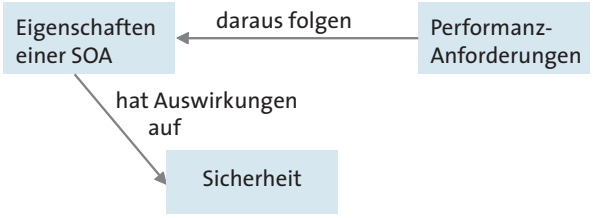


Abbildung 55: Auswirkungen der Performanz auf die Sicherheit

zu (1): Auswirkungen der SOA-Sicherheit auf die Performanz:

- Verteilte Systeme
  - SOA-Eigenschaft:
    In einem SOA-System sind die Funktionen und Services auf getrennte Komponenten verteilt.
  - Daraus resultierende Sicherheitsanforderung:
    Die Verteilung führt zu einem hohen Bedarf an sicherer Kommunikation zwischen diesen Komponenten. Eine sichere Kommunikation kann im Falle eines nicht vertrauenswürdigen Kommunikationskanals bedeuten, dass die komplette Kommunikation verschlüsselt werden muss und dass sich alle Kommunikationspartner gegenseitig authentisieren müssen.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Verschlüsselung und Authentisierung führen häufig neben einer Steigerung der benötigten Rechenleistung auch zu einer erhöhten Netzwerklast (durch Metadaten) und einer Reduzierung der übertragbaren Nutzdaten pro Zeiteinheit. Dies trifft insbesondere auf stark verteilte Systeme zu. Die Folge können vor allem bei mobilen Endgeräten Performanz-Probleme sein bzw. zu Kompromissen in Form von mangelhafter Sicherheit führen.
  - Lösungsansatz:
    Der gesteigerten Anforderungen an die Rechenleistung kann mit speziellen Krypto-Karten begegnet werden, die entsprechende Rechenkapazitäten zur Verfügung stellen. Dies hat aber insbesondere bei mobilen Endgeräten den Nachteil des höheren Energieverbrauchs.
    Ein anderer Ansatz besteht darin, die Verteilung des Systems zu verringern, indem die Services redundant auf verteilter Hardware angeboten werden (Load Balancing).
    Anstelle der Verschlüsselung der Kommunikation kann auch der Kommunikationskanal selbst gesichert werden. Eine Authentisierung ist aber weiterhin notwendig.

- Lose Koppelung
  - SOA-Eigenschaft:
    Ein SOA-System ist von einer hohen Dynamik und Flexibilität geprägt. Viele verschiedene Instanzen kommunizieren miteinander. Das Zusammenwirken von Services wird zum Teil erst während der Laufzeit bestimmt.
  - Daraus resultierende Sicherheitsanforderung:
    Um gesichert zwischen den Komponenten kommunizieren zu können, sind viele dynamische Vertrauensbeziehungen notwendig.
    Dies bedeutet häufiges neues Aufbauen dieser Vertrauensbeziehungen, wobei sich die Rollen der Services ändern können. Die Bildung einer solchen Vertrauensbeziehung beginnt dabei in der Regel mit einer Identitätsfeststellung (Authentifizierung). Authentifizierungen von SOA-Komponenten erfolgen in der Regel per Zertifikat. Zertifikate müssen dabei auf ihre Gültigkeit hin überprüft werden.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Dies bedeutet erhöhte Rechenleistung sowohl bei demjenigen, der sich authentisieren muss, als auch beim Authentisierer. Hinzu können Anfragen an die ausstellende Certificate Authority (wegen CRLs bzw. OCSP) oder andere Trusted Third Parties (Authentication/Authorization Authority) kommen, was wiederum den Prozesses zeitlich erheblich belastet.
  - Lösungsansatz:
    Abhilfe bieten hier lokal in die Nähe der SOA-Komponenten verteilte Certificate Authorities bzw. OCSP-Responder/CRL-Directories. Generell gilt, dass Zertifikatsketten möglichst kurz gehalten werden sollten. Wo möglich, sollten einmal etablierte vertrauenswürdige Identifikationen mittels eines Tunnels aufrecht erhalten werden. Im Falle zweier Endpunkte bietet sich hierzu SSL an, dabei erfolgt die Authentisierung aber nur auf Netzwerkebene, nicht auf Applikationsebene. SSL kann im

Zusammenspiel mit instanziierten Services oder mit Load Balancing zu Problemen führen.

- Verteiltes System (2)
  - SOA-Eigenschaft: s.o.
  - Daraus resultierende Sicherheitsanforderung:
    Ein weiterer Aspekt von verteilten Systemen ist, dass es notwendig ist, die korrekte Übermittlung (im Sinne einer Transaktion) zu bestätigen. Gegebenenfalls muss in komplexen Orchestrierungen (d. h. dem Zusammenwirken mehrerer Funktionen bzw. Services) nicht nur der Status einer jeden Transaktion gehalten werden, sondern auch rückabgewickelt werden können. Eine Rückabwicklung kann erforderlich werden, wenn während der Ausführung eines Prozesses eine Unterbrechung eintritt, z. B. wegen Netzwerkfehlern oder wegen fehlender Zugriffsberechtigungen.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Aufwand durch Session-/Statusinformationen
  - Lösungsansatz:
    Wird die Orchestrierung von Services durch eine Zentrale ausgeführt, kann diese die Steuerung von Sessions effizient durchführen.
    Session-/Statusinformationen auf Applikationsebene sind hingegen von den einzelnen Services zu verwalten.

- Wiederverwertbarkeit und Orchestrierung
  - SOA-Eigenschaft:
    Einzelne SOA-Services werden über die „Orchestrierung“ zu Service-Abfolgen (Prozessketten) oder auch neuen, zusammengesetzten SOA-Services verbunden. Erklärtes Ziel von SOA ist es, die Wiederverwertbarkeit von einzelnen Funktionen und Services zu steigern.
  - Daraus resultierende Sicherheitsanforderung:
    Werden SOA-Funktionen/-Services in verschiedenen Prozessen bzw. Services eingesetzt, so kann sich ihr Sicherheitskontext ändern. Soll also eine SOA-Funktion z. B. in zwei Services Verwendung finden, muss die Funktion zwangsweise die jeweils höheren Anforderungen dieser Services hinsichtlich Verfügbarkeit, Vertraulichkeit, Integrität und Verbindlichkeit unterstützen.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Bei Einhaltung des Maximierungsprinzips werden hohe Sicherheitsanforderungen einzelner Services immer, d. h. bei allen Services, auf die beteiligten Funktionen angewendet bzw. müssen in den Services abrufbar sein – nicht nur bei den Services, die dies tatsächlich erfordern. Dadurch werden unnötig hohe Sicherheitsanforderungen an die Funktionen gestellt, die nur in Einzelfällen notwendig wären. Die Erfüllung dieser hohen Sicherheitsanforderungen hat im Allgemeinen negative Auswirkungen auf die Performanz.
  - Lösungsansatz:
    Um die Anforderungen an einzelne SOA-Funktionen zu begrenzen, sind sinnvolle Cluster von Services (Security-Domänen) zu bilden, die ähnliche Anforderungen an Verfügbarkeit, Vertraulichkeit, Integrität und Verbindlichkeit haben. Falls nötig, kann ein Service in unterschiedlichen Security-Domänen mehrfach instanziiert oder implementiert werden.
    Da sich aufgrund der hohen Dynamik einer SOA-Landschaft diese Anforderungen rasch ändern können, ist ein stetiges Monitoring der Nutzung der SOA-Funktionen zwingend erforderlich. Dieses Monitoring ergänzt jedoch nur ein ausgereiftes Change Management für SOA-Services (Service Lifecycle Management).

- Komplexe Semantik und babylonische Sprachvielfalt
  - SOA-Eigenschaft:
    Eine der Stärken von SOA beim Einsatz von XML-basierten Schnittstellen ist deren Flexibilität bei der Definition der Service-Schnittstellen, etwa verglichen mit den sehr unflexiblen Definitionen in der EDI-Welt (EDIFACT). So müssen formale syntaktische Spezifikationen nicht immer vollständig sein; zudem entspricht die Toleranz gegenüber syntaktisch unbekannten, aber formal korrekten Formulierungen der XML-Philosophie. Die entstehenden Schnittstellen-Konstruktionen

können dadurch sehr komplex und unübersichtlich werden.

- Daraus resultierende Sicherheitsanforderung:
  Umfangreiche und vor allem vollständige formale Prüfungs-Routinen sind nötig, um die übermittelten Informationen korrekt auswerten zu können. Diese Möglichkeiten (content-based security) sind grundsätzlich gegeben, müssen aber auch entsprechend eingesetzt werden.
  Zudem entstehen Sicherheitsprobleme aus gezielt missgestalteten Nachrichten (zum Beispiel mit zu tiefer Rekursion oder gezielten Low-Level-Format-Fehlern). Hierauf müssen die Daten vor der Verarbeitung entsprechend untersucht werden.
- Daraus resultierende Auswirkungen auf die Performanz:
  Die notwendigen Prüfungen verursachen einen nicht unerheblichen Zeitaufwand und eine ebensolche Rechenlast.
- Lösungsansatz:
  Man sollte sich auf ein möglichst einfaches und kleines Subset der zur Verfügung stehenden Möglichkeiten zur Definition von XML-basierten Service-Schnittstellen beschränken. Dabei ist es wichtig, Schnittstellen-Definitionen möglicht vollständig zu formulieren (eine Governance-Aufgabe) und auf Stabilität der eingesetzten Prüfwerkzeuge (XML-Parser) zu achten, um damit möglichen Angriffen des Typs „denial of service“ zu widerstehen. In Teilen kann man die Last auch durch mehrfache Instanziierung und Load Balancing verteilen. Weiter stellen spezialisierte XML-Appliances (zum Beispiel IBM DataPower) für derartige hochperformante und stabile Prüfungsaufgaben einen möglichen Ansatz zur Verfügung.

- Verwendung der Funktionen/Services durch viele Instanzen
  - SOA-Eigenschaft:
    SOA-Funktionen/-Services stehen grundsätzlich jedem zur Verfügung und können von vielen verschiedenen Instanzen (Service Consumer) verwendet werden.
  - Daraus resultierende Sicherheitsanforderung:
    Es ist daher schwierig nachzuvollziehen, wer wann welche Funktion oder welchen Service aufgerufen hat. Es entstehen erhöhte Logging-Anforderungen, um Fehler oder Missbrauch nachvollziehen zu können. Gegebenenfalls entstehen zusätzlich Anforderungen für ein Accounting bei kostenpflichtigen Funktionen/Services. Diese Anforderungen sind allerdings Auswirkungen neuer Betriebs- und Verrechnungskonzepte, die immer auftreten, wenn solche flexiblen Betriebs- und Verrechnungskonzepte eingesetzt werden sollen, auch unabhängig von einer SOA. SOA ist hier nur ein Ansatz, der solche Betriebsmodelle fördert.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Die Performanz wird negativ beeinflusst durch erhöhte Rechenleistung, erhöhtes Übertragungsvolumen (Netzlast), erhöhten Speicherplatz und durch Aufwände zur Sicherstellung der Sicherheit der Loggingdaten selbst.
  - Lösungsansatz:
    In einem Logging-Konzept ist zu definieren:
    - Welche Daten werden geloggt?
    - Bei welchem Ereignis werden sie geloggt?
    - Mit welchem Ziel werden sie geloggt?
    - An welches System werden sie gesendet und wo sind sie gespeichert?
    - Wie werden die Daten ausgewertet?
    - Die erhöhten Logging-Anforderungen, auch für flexible Nutzungskonzepte, sind von vornherein einzuplanen.

- Verwendung der Funktionen/Services durch viele Instanzen (2)
  - SOA-Eigenschaft: s.o.
  - Daraus resultierende Sicherheitsanforderung:
    Funktionen/Services, die nicht öffentlich jedem zur Verfügung stehen, müssen vor ihrer Ausführung die entsprechenden Zugriffsrechte überprüfen. Dafür sind die in den vorigen Abschnitten beschriebenen Komponenten PEP, PDP und PAP notwendig.

- Daraus resultierende Auswirkungen auf die Performanz:
  PEP/PDP
  Die Überprüfung der Zugriffsrechte ist integraler Bestandteil der über die SOA abgewickelten Transaktionen. Damit hat die Performanz dieser Security-Operationen direkten Einfluss auf die Gesamtperformanz der SOA-Transaktionen. Die Security-Operationen müssen entsprechend schnell abgewickelt werden. Dies umfasst die Performanz der Übertragung und der Rechenoperation, insbesondere der PEP- und PDP-Komponenten. Kennzeichnend für Operationen dieser Art sind ein hohes Anfragevolumen bei eher geringerem Datenvolumen.
  PAP
  Die PAP-Komponenten werden für die Verwaltungsfunktionen verwendet und haben deshalb niedrigere Anforderungen an die Performanz. Die Verwaltungsoperationen zeichnen sich durch ein eher geringeres Anfragevolumen bei größeren Datenmengen aus.
- Lösungsansatz:
  - PEP/PDP:
    Daten nahe der Komponenten
    Load Balancing
    Geringe Netzwerk-Latenz-Zeiten
    Aufgeteilte Datenbasis
    Konsequente funktionale Aufgabenteilung (Redundanzvermeidung)
    Hardware-Beschleunigung (Crypto)
  - PAP:
    Gute Volumenverarbeitungseigenschaften
    Guter Netzwerkdurchsatz

- Federation
  - SOA-Eigenschaft:
    In einer SOA-Umgebung können die Identitäten und Rollen der beteiligten Entitäten oft in verschiedenen Identitätsmanagement-Systemen (Identity Mangament Systems) verwaltet werden. Auch hier gilt wieder: Dieses teilweise unabgestimmt redundante Identitätsmanagement kann auch ohne SOA in komplexeren Applikation-Integrations-Aufgabenstellungen resultieren. SOA kann diese bestehenden Schwächen allerdings noch deutlicher hervorheben.
  - Daraus resultierende Sicherheitsanforderung:
    Bei der Verwendung der Identitäten und Rollen für Authentisierung oder Autorisierung müssen Identitäten der verschiedenen IDMs sicher „gemappt" werden können.
  - Daraus resultierende Auswirkungen auf die Performanz:
    Die Abfrage und Zusammenführung von Identitäten und Rollen bei verschiedenen IDMs ist zeitaufwendig.
  - Lösungsansatz:
    Das Mappen von Identitäten ist eine klassische Aufgabe von IAM-Lösungen oder Meta-Directories. SAML bietet Lösungen für Federated Identity.

zu (2): Auswirkungen der Performanz auf die Sicherheit

- Performanz-Anforderung:
  Im Rahmen der Authentisierung soll bei der Prüfung von Zertifikaten nicht die Prüfung des gesamten Zertifikatsbaums notwendig sein.

- Daraus folgende SOA-Lösung:
  Es ist deutlich performanter, mit einem Proxy-User zu arbeiten, der zum Beispiel Teil des ESB sein kann. Der ESB agiert dabei im Namen („on behalf") von SOA-Komponenten und Nutzern, die ihm gegenüber bereits authentifiziert und autorisiert sind. Die zeitaufwendige Überprüfung von zum Beispiel Zertifikatsketten entfällt weitgehend bzw. erfolgt nur einmalig gegenüber dem ESB.

- Auswirkungen auf die Sicherheit:
  Proxy-User anonymisieren eine Verbindung und agieren „on behalf". Ein ESB, der als Proxy arbeitet, kann zu einem erheblichen Vertrauensverlust führen, da letztlich der ESB immer sagt „es kommt doch von mir – vertrau mir" und damit einen „single point of failure" darstellt. Die alleinige Vertrauensinstanz ist damit der ESB.
  Die Sicherheit einer SOA-Lösung wird durch den

Einsatz von Proxy-Usern auf das Niveau klassischer Legacy-Systeme zurückgeführt, aber in einem potenziell veränderten Geschäftskontext (z. B. externe Partner). Zumindest müssen die daraus erwachsenden Implikationen für den Schutzbedarf genau aufgearbeitet werden.
Auf der anderen Seite wird die Sicherheit zentral an einer Stelle gebündelt und kann somit leichter zu überwachen und zu administrieren sein. Die Anforderungen an die Sicherheit des ESB wachsen bei diesem Lösungsansatz dementsprechend.
Ebenso ist zu beachten, dass trotz des Einsatzes eines Proxy-Users in den Log-Files eine vollständige Nachvollziehbarkeit der gesamten Transaktion gegeben sein muss (End-to-End).

Verfügbarkeit

Authentisierung, Autorisierung und Verschlüsselung sind Schlüsselfunktionen, deren Verfügbarkeit unabdingbar für den Betrieb der gesamten SOA ist. Dadurch ergeben sich hohe Verfügbarkeitsanforderungen an die ausführenden Komponenten. Im Gegensatz zu diesen Komponenten, die zur Laufzeit funktionieren müssen, gibt es Policy-Administration-Punkte (PAP), die zur Verwaltung und Konfiguration Security-relevanter Informationen dienen. Auch diese Funktionalität wird für den Betrieb der SOA-Infrastruktur grundsätzlich benötigt, allerdings sind hier die Anforderungen an die Verfügbarkeit andere, zumeist geringere.

Komponenten	Klassifizierung	Maßnahmen
Policy-Enforcement- und Policy-Decision-Punkte	hohe Verfügbarkeit	■ Datenreplikationen nahe der Komponenten ■ Mehrfachauslegung und automatische Umschaltung ■ Hochverfügbarkeit ■ Aktive Systemüberwachung
Policy-Administration-Punkte	mittlere Verfügbarkeit	■ Geeignete Back-up-Verfahren ■ Separierung aus Netzwerksicht (Managementzone) ■ Redundante Netzwerkanbindung

Security

Auch für die relevanten Komponenten der SOA-Security ist auf eine angemessene und sichere Ausgestaltung zu achten („Sicherheit der Security“).

Für den Bereich der Policy-Enforcement- und Policy-Decision-Punkte ist vor allem sicherzustellen, dass die Integrität der Komponenten und Verwaltungsdaten gewährleistet ist und nicht kompromittiert werden kann. Da diese Komponenten oft – aus Gefährdungssicht – in exponierter Lage angesiedelt sind, sind die Systeme, auf denen diese Komponenten installiert werden, angemessen zu konfigurieren. Des Weiteren muss die Anbindung an die „Security-Verwaltungskomponenten“ sicher ausgestaltet sein.

Die Policy-Administration-Punkte werden typischerweise aus Netzwerksicht in einer eigenen Administrationszone angeordnet und sind damit vom restlichen Netz separiert. Der Zugang zu diesen Systemen, der primär Verwaltungszwecken dient, sollte nur bestimmten Benutzerkreisen (Administratoren) erlaubt sein. Eine stärkere Authentifizierung (z. B. 2-Faktor-Autorisierung) ist in diesem Bereich angebracht.

Komponenten	Klassifizierung	Maßnahmen
Policy-Enforcement- und Policy-Decision-Punkte	hohe Sicherheit	■ Gehärtete und dedizierte Systeme ■ Gesicherte Anbindung an Management-Komponenten ■ Audit Logs ■ Aktives Patch Management
Policy-Administration-Punkte	hohe Sicherheit	■ Separate Netzwerkzone ■ Starke Authentifizierung ■ Gehärtete Systeme ■ Audit Logs

Randbemerkungen zur Praxis

An dieser Stelle sei noch auf zwei Problembereiche aus der Praxis im Umgang mit SOA-Infrastrukturen hingewiesen:

Da SOA-Infrastrukturen häufig teilweise von einem Team von Architekten, Entwicklern und Herstellern konzipiert und zusammengestellt werden, geht schnell der Blick für das Ganze verloren. Dies führt dazu, dass die oben beschriebenen Kriterien nicht ausreichend berücksichtigt werden, was im Praxiseinsatz zu Problemen führt. Aus diesem Grund ist es unerlässlich, alle oben angesprochenen Probleme hinsichtlich Performanz, Verfügbarkeit und Sicherheit bereits zu Beginn eines Projektes in die Gesamtprojektplanung einzubeziehen.

Da eine größere SOA häufig aus einer ganzen Reihe von verteilten Komponenten besteht, ist eine einfache Ursachenanalyse im Bereich Performanz im Problemfall oft nicht möglich. Hier hat sich in der Praxis der Einsatz einer „Application-Response-Time“-Analyse im Problemfall sehr bewährt.

8 Standards (Steckbriefe)

8.1 Technische Standards

Web Service-Universum

Einer der wichtigsten Standards im SOA-Umfeld ist der Web Services Standard. Der setzt sich wiederum aus zahlreichen Standards zusammen („Web Service Universum"). In diesem Kapitel wird ein Überblick über das „Web Service-Universum" gegeben und die Technologien erklärt mit Verweisen auf Quellen zur weiteren Recherche. Außerdem bewertet es den aktuellen Reifegrad bzw. die momentane Verbreitung.

Unter „http://www.innoq.com/resources/ws-standards-poster" findet man ein Poster mit einem vollständigen Überblick über die Web Service-Standards zum Download.

Die wichtige Standardisierungsgremien sind:

- W3C
- OASIS
- WS-I
- Java Community Process (JCP)

Alle Standards zu beschreiben, würde den Rahmen dieses Kapitels und den des zu erstellenden Dokuments sprengen. Außerdem ist es nicht möglich, so ein Dokument ständig zu aktualisieren.

Ziel ist es also:

- einen größtmöglichen Überblick über die existierenden Standards zu schaffen.
- einen entsprechenden Zusammenhang zu den anderen Standards herzustellen.
- kurz zu erklären, wofür jeder Standard steht.
- die Akzeptanz bzw. den aktuellen Status des Standards darzustellen.
- einen schnellen Verweis auf die Spezifikation bzw. gute weitere Quellen zu liefern.

Um die Pflege leichter und übersichtlich zu gestalten, sind die Standards in Steckbriefen erfasst und thematisch zu Blöcken zusammengefasst.

Web Service-Basistechnologien

SOAP					
Kurzbeschreibung	Das Simple Object Access Protocol (SOAP) ist ein einfaches XML basiertes Protokoll zum Austausch von Informationen in verteilten Systemen. Unabhängig vom Transportprotkoll unterstützt es dabei sowohl dokumentenbasierten (datenorientierten / „dokument style"), als auch den bekannteren funktionsbasierten („RPC style") Austausch von Informationen.				
Besitzer	W3C				
Version	1.2	Datum	27.07.2007	Status	Recommendation
Version Nachfolger		Datum		Status	
Zusammenhang zu anderen Standards	SOAP definiert das Format für Nachrichten (auch für Funktionsaufrufe). Die inhaltliche und formelle Struktur spezieller SOAP Nachrichten für einen WebService werden in dessen Beschreibung (WSDL) definiert und konkreten Operationen zugeordnet. Für zusätzliche Informationen definiert SOAP innerhalb des Envelopes einen Header, in dem z.B. Informationen zur Sicherheit oder Addressierung formuliert werden können. Hier gibt es weitere Standards wie WS-Adressing.				

Gruppe	Web Service Basistechnologien
Akzeptanz	allgemein akzeptiert
Spezifikation / Quelle	z.B. http://www.w3.org/TR/2007/REC-soap12-part0-20070427/
Weitere Links / Quellen	z.B. http://de.wikipedia.org/wiki/SOAP

WSDL

Kurzbeschreibung	Die Web Service Description Language (WSDL) definiert den formellen Inhalt, den Zugriff und das Protokoll eines auf WebService Technologie basierten Dienstes (Binding). Dabei werden abstrakte Operationen eines Dienstes in Nachrichten inkl. Richtung und Reihenfolge zerlegt. Die Nachrichten sind inhaltlich so spezifiziert, dass sich daraus die genaue SOAP Syntax ergibt („das XML Schemas für SOAP Nachrichten“ im weiteren Sinne. Außerdem wird die konkrete Bindung zu einem Protokoll inkl. Adresse (zu den einzelnen Endpunkten) für den entsprechenden Dienst definiert. Die Beschreibungen werden typischer Weise in zentralen Repositories gesammelt und verwaltet.				
Besitzer	W3C				
Version	1.1	Datum	15.03.2001	Status	Note
Version Nachfolger	2.0	Datum		Status	Working Draft
Zusammenhang zu anderen Standards	Nachrichtenformat (SOAP), Repository Zugriff (UDDI)				
Gruppe	Web Service Basistechnologien				
Akzeptanz	allgemein akzeptiert				
Spezifikation / Quelle	http://www.w3.org/TR/wsdl.html				
Weitere Links / Quellen	http://de.wikipedia.org/wiki/WSDL				

UDDI

Kurzbeschreibung	Die Universal Description, Discovery and Integration (UDDI) definiert die Schnittstelle, die ein Verzeichnisdienst zur Verwaltung und zum Auffinden bereitstellt. Es werden oft Vergleiche zum Telefonbuch („White Pages“), den gelben Seiten („Yellow Pages“) und dem Unternehmensmodell („Green Pages“) hergestellt. Der Zugriff erfolgt via SOAP.				
Besitzer	OASIS				
Version	2.0	Datum	19.07.2002	Status	Published
Version Nachfolger	3.0	Datum		Status	Committee Draft
Zusammenhang zu anderen Standards	Nachrichtenformat (SOAP), Repository Zugriff (UDDI), Beschreibung der Dienste (WSDL)				
Gruppe	Web Service Basistechnologien				
Akzeptanz	akzeptiert, aber noch wenig eingesetzt				
Spezifikation / Quelle	http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm				
Weitere Links / Quellen	http://de.wikipedia.org/wiki/UDDI				

Web Services der zweiten Generation

SAML					
Kurzbeschreibung	Die Security Assertion Markup Language (SAML) definiert den Austausch von Authentifizierungs- und Autorisierungsdaten zwischen einem Identity Provider und einem Service-Provider. Mit Hilfe von SAML läßt sich eine SingleSignOn (SSO) Lösung aufbauen, die über Unternehmensgrenzen hinweggeht. Daten werden in Form von Assertions als XML Dokumente ausgetauscht. Der Identity Provider stellt Zusicherungen (Assertions) aus und der Service-Provider verwendet diese. Weitere Bestandteile der Spezifikation sind das SAML-Protokoll, die SAML-Bindings und die Profile.				
Besitzer	OASIS				
Version	1.0	Datum	November 2002	Status	Standard
Version Nachfolger	1.1	Datum	September 2003	Status	Standard
Version Nachfolger	2.0	Datum	März 2005	Status	Standard
Zusammenhang zu anderen Standards	Sicherheit (WS-Security), Single Sign On (Liberty) Verschlüsselung (XML Encryption)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	allgemein akzeptiert				
Spezifikation / Quelle	http://www.oasis-open.org/committees/security/				
Weitere Links / Quellen	http://de.wikipedia.org/wiki/SAML				

WS - Addressing					
Kurzbeschreibung	Definition eines SOAP Headers bzw. Mechanismus, um Web Services transportneutral zu addressieren.				
Besitzer	W3C				
Version	1.0	Datum	09.05.2006	Status	Recommendation
Version Nachfolger		Datum		Status	
Zusammenhang zu anderen Standards	Protokoll-Bindung in der Web Service Beschreibung (WSDL) Headerintegration (SOAP)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	noch relativ unbekannt				
Spezifikation / Quelle	http://www.w3.org/2002/ws/addr/				
Weitere Links / Quellen					

WS - MetadataExchange

Kurzbeschreibung	Ein Webdienst beschreibt über Metadaten die Eigenschaften, die ein Benutzer kennen muss, um den Webdienst zu verwenden. Beispiele für Metadaten sind: ■ Policies: WS-Policy [WS-Policy] beschreibt die Eigenschaften, Anfoderungen und allgemeinen Charakteristika eines Webdienstes ■ WSDL [WSDL 1.1] beschreibt die abstrakten Operationen, konkretes Netzwerk-Protokoll, die Endpunkt-Adressen die vom Webdienst verwendet werden. ■ XML Schema [XML Schema Part 1, Part 2] beschreibt die Struktur und den Inhalt der XML Nachrichten die vom Webdienst erhalten und versendet werden. Um die Kommunikation mit Webdiensten "anzustarten" (bootstrap) wird in dieser Spezifikation festgelegt, wie Metadaten ausgetauscht werden. Weiterhin wird festgelegt wie Metadaten in einen Web Service Endpunkt eingebettet werden und wie Web Services Endpunkte optional selbst eine Request-Reply Interaktion unterstützen können, um Metadaten zur Verfügung zu stellen.				
Besitzer	Aktuelle Version nicht bei einem Standardgremium veröffentlicht.				
Version		Datum	August 2006	Status	Public Draft Release
Version Nachfolger		Datum		Status	
Zusammenhang zu anderen Standards	Policy (WS-Policy), Transfer (WS-Transfer), Schnittstelle (WSDL)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	dabei sich zu etablieren				
Spezifikation / Quelle	http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf				
Weitere Links / Quellen	http://en.wikipedia.org/wiki/WS-MetadataExchange				

WS - ReliableMessaging

Kurzbeschreibung	Bei der Übermittlung von Nachrichten können vielfältige Fehler auftreten. Nachrichten können verloren gehen, doppelt gesendet werden oder in der Reihenfolge vertauscht werden. Um eine Nachricht zuverlässig vom einem Sender (Application Source (AS)) zu einem Empfänger, über eine unzuverlässige Verbindung, zu übermitteln geht der Sender wie folgt vor: Sender und Empfänger bedienen sich dazu Zwischenstellen (Reliable Messaging Source (RMS), Reliable Messaging Destination (RMD)) die sicherstellen, das eine Nachricht mit der zugesicherten Garantie übertragen wird. Ein Sender übermittel eine Nachricht an den RMS. Dieser kommuniziert zuverlässig über das WS-ReliableMessaging (WS-RM) Protokoll mit dem RMD. Der RMD stellt die Nachricht an den Empfänger zu. Kann ein RMS die Nachricht nicht an einen RMD übermitteln, so muss dies als Fehler an den Sender weitergegeben werden. Als Garantien für eine Nachrichten-Übermittlung unterstützt: ■ AtLeastOnce – eine Nachricht wird zumindest einmal zugestellt. Nachrichten können jedoch mehrmals zugestellt werden. ■ AtMostOnce – eine Nachricht wird höchstens einmal zugestellt. Nachrichten werden nie zweimal zugestellt, können jedoch auch gar nicht zugestellt werden. ■ ExactlyOnce - eine Nachricht wird genau einmal zugestellt. ■ InOrder – Nachrichten werden in der Reihenfolge empfangen in der die Nachrichten gesendet wurden.

Besitzer	OASIS				
Version	1.1	Datum	14.6.2007	Status	Veröffentlicht
Version Nachfolger		Datum		Status	
Zusammenhang zu anderen Standards	Sicherheit (WS-Security), Policy (WS-Policy)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	allgemein akzeptiert				
Spezifikation / Quelle	http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.1-spec-cs-01.pdf				
Weitere Links / Quellen	http://en.wikipedia.org/wiki/WS-ReliableMessaging				

WS - SecureConversation

Kurzbeschreibung	WS-Security bietet Mechanismen für das Sichern einer einzelnen Nachricht bei einem Nachrichtenaustausch. Interaktionen zwischen einem Webdienst und einem Benutzer führen jedoch meist zum Austausch mehrerer Nachrichten. Obwohl jede Nachricht einzeln gesichert werden kann, ist es doch effizienter, einen vom Webdienst und dem Benutzer gemeinsam verwendeten Kontext zu schaffen und mit diesem, die durch das Sichern jeder ausgetauschten Nachricht, entstehende Arbeitslast zu reduzieren. WS-SecureConversation definiert ein Sicherheitskontext-Token (Security Context Token), das diese Aufgabe ausführt.				
Besitzer	Aktuelle Version nicht bei einem Standardgremium veröffentlicht. Nachfolgeversion wird bei der OASIS standardisiert.				
Version		Datum	November 2002	Status	Public Draft Release
Version Nachfolger		Datum		Status	
Zusammenhang zu anderen Standards	Sicherheit (WS-Security), Sicherheit (WS-Trust)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	dabei sich zu etablieren				
Spezifikation / Quelle	Aktuelle Version: http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf Zukünftige Version: http://www.oasis-open.org/committees/ws-sx/charter.php				
Weitere Links / Quellen	http://en.wikipedia.org/wiki/Web_Services_Security http://en.wikipedia.org/wiki/Web_Services_Interoperability_Technology				

WS - Security

Kurzbeschreibung	Die WS-Security Spezifikation legt fest, wie eine SOAP Nachricht aufgebaut sein muss, damit die Vertraulichkeit und die Integrität sichergestellt sind. Dazu werden folgende SOAP Header festgelegt: a) Security Token, b) Unterschrift. Um die Vertraulichkeit einer SOAP Nachricht zu erreichen, können Nachrichtenteile (Header,Body) verschlüsselt werden. Um die Integrität einer SOAP Nachricht sicherzustellen, werden die Nachrichtteile (Header,Body) unterschrieben.

Besitzer	OASIS				
Version	1.0	Datum	19. April 2004	Status	Standard
Version Nachfolger	1.1	Datum	17. Februar 2006	Status	Standard
Zusammenhang zu anderen Standards	Sicherheit (WS-SecureConversation), Verschlüsselung (XML Encryption), Signatur (XML Signature)				
Gruppe	Web Service der zweiten Generation				
Akzeptanz	allgemein akzeptiert				
Spezifikation / Quelle	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss				
Weitere Links / Quellen	http://en.wikipedia.org/wiki/Web_Services_Security				

Java

JBI (JSR 208)					
Kurzbeschreibung:	Das Ziel der JBI-Spezifikation besteht in einer Vereinfachung von Integrationsprojekten über Plattformgrenzen hinweg. JBI beschreibt eine auf Web-Service-Standards basierende erweiterbare Integrationsarchitektur, die aus Containern und Komponenten besteht. Container stellen die Laufzeitumgebung für die Komponenten bereit, die über ein abstraktes Service-Modell miteinander kommunizieren. JBI beschreibt eine SOA-Rahmenarchitektur, in der die Komponenten als Dienst-Anbieter und Dienst-Nutzer betrachtet werden und Mechanismen zur Realisierung von Enterprise Service Busses. Die JBI-Laufzeitumgebung besteht aus einer Anzahl von Komponenten, die innerhalb einer Java Virtual Machine ablaufen. Dienste können sowohl außerhalb der JBI-Umgebung erbracht und an diese durch spezielle Bindungskomponenten (BC- binding components) angeschlossen werden oder sie können in Containern - so genannten service engines - innerhalb der JBI-Umgebung realisiert werden. Beispiele für service-engines sind EJB-Container, WS-BPEL-Container, Container für Regelbearbeitungen oder JSP/Servlet-Container. An eine JBI-Umgebung angeschlossene Dienste kommunizieren über einen normalized message router, einen Dienst der von der JBI-Umgebung angeboten wird. Obwohl der Standard bereits im August 2005 publiziert wurde, existieren nur wenige kommerzielle und Open Source Implementierungen der JBI-Spezifikation. Der Hintergrund dafür ist sowohl in der Komplexität des Standards selbst als auch in der Schwierigkeit zu sehen, die Spezifikation umzusetzen.				
Besitzer:	SUN				
Version:		Datum:	08/2005	Status:	Final
Zusammenhang zu anderen Standards:	Sowohl für die Spezifikation des abstrakten Kommunikationsmodells als auch für die Bindung an eine existierende Technologie wird die Web Service Description Language (WSDL) V2 verwendet Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft 26 March 2007 http://www.w3.org/TR/wsdl20/				

Gruppe:	Java
Akzeptanz:	akzeptiert, aber noch wenig eingesetzter Standard
Spezifikation / Quelle:	http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html
Weitere Links / Quellen:	

XML Basistechnologie

XML & DTD

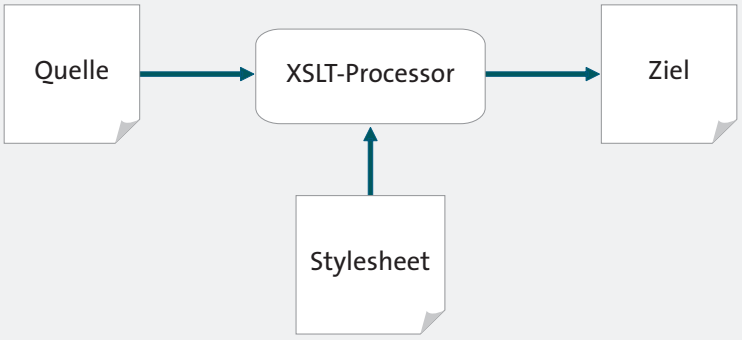
Kurzbeschreibung:	Die Extensible Markup Language (XML) ist eine Auszeichnungssprache für hierarchisch strukturierte Texte. Ursprünglich wurde XML für die Beschreibung und den Austausch von Texten konzipiert, welche für einen menschlichen Leser gut verständlich sein sollten (= document centric). Diese ursprünglich Ausrichtung hat sich mittlerweile in Richtung eines maschinellen Datenaustausches (= data centric) verschoben. Ein XML-Dokument besteht aus einzelnen Elementen, welche beliebig verschachtelt werden können, wobei genau ein Wurzelelement existieren muss. In der Darstellung als Textdokument werden Elemente durch so genannte Tags begrenzt, welche den Beginn (Start-Tag: <Element>) und das Ende (End-Tag: </Element>) definieren. Unter der Voraussetzung, dass Elemente korrekt ineinander verschachtelt sind (Elemente dürfen sich nicht überschneiden), wird ein XML-Dokument als „well-formed“ bezeichnet. Die Struktur (Grammatik) von XML-Dokumenten kann über eine so genannte Document Type Definition (DTD) beschrieben werden. Eine DTD definiert den Aufbau jedes einzelnen Elements, indem die Struktur der Unterelemente definiert wird. Der Inhalt eines Elementes setzt sich aus weiteren Elementen, Text oder einer Mischform aus beidem zusammen. Ein XML-Dokument wird als „valid“ bezeichnet, wenn die Dokumentstruktur der in der DTD beschriebenen Syntax entspricht. Diese Prüfung kann automatisch durch einen Parser erfolgen. Aufgrund seiner ursprünglichen Definition als „Textformat“ hat die DTD als Beschreibungsmittel eine beschränkte Ausdruckskraft. Dies äußert sich z.B. dadurch, dass keine Unterscheidung zwischen verschiedenen Datentypen (Zahl, Datum, String, ...) möglich ist.				
Besitzer:	W3C				
Version:	1.0, Fourth Edition	Datum:	16.08.2006	Status:	Recommendation
Version Nachfolger:		Datum:		Status:	
Zusammenhang zu anderen Standards:	XML ist Ausgangspunkt vieler weiterer Standards, welche XML entweder direkt als Datenformat verwenden, oder weitere Anwendungen direkt auf Basis von XML definieren. Beispiele hierfür sind: XPATH, XSLT, XML-Schema, WSDL, SOAP.				
Gruppe:	XML Basistechnologie				
Akzeptanz:	allgemein akzeptiert				
Spezifikation / Quelle:	http://www.w3.org/TR/2006/REC-xml-20060816/				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/XML				

XML-Schema					
Kurzbeschreibung:	XML-Schema ist ein Hilfsmittel um die Struktur von XML-Dokumenten auf Basis eines Typsystems zu beschreiben. XML-Schema nimmt eine Trennung zwischen Elementen und Typen vor. Das Konzept „Typ“ steht hierbei für eine Strukturdefinition, welche ein Fragment eines XML-Dokuments beschreibt. Grundsätzlich wird zwischen einfachen und komplexen Typen unterschieden. Einfache Typen sind z.B. Zahlen, Datum, String, Boolean. Komplexe Typen werden verwendet, um Elemente zu beschreiben, welche weitere Kindelemente und Attribute enthalten. XML-Schema geht weit über den Umfang einer DTD hinaus und erlaubt die exakte Beschreibung sowohl von daten-, als auch von dokument-zentrierten XML-Dokumenten. Als Schemasprache findet XML-Schema sehr breite Anwendung welche zunehmend die Rolle von Document Type Definitions übernimmt.				
Besitzer:	W3C				
Version:	1.1	Datum:	18.10.2004	Status:	Recommendation
Version Nachfolger:		Datum:		Status:	
Zusammenhang zu anderen Standards:	XML-Schema ist ein weit verbreiteter Standard welche von vielen Standards verwendet wird, um eigne XML-Strukturen zu definieren..				
Gruppe:	XML Basistechnologie				
Akzeptanz:	allgemein akzeptiert				
Spezifikation / Quelle:	Part 0: Primer: http://www.w3.org/TR/xmlschema-0/ Part 1: Structures: http://www.w3.org/TR/xmlschema-1/ Part 2: Datatypes: http://www.w3.org/TR/xmlschema-ref/				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/XML-Schema http://www.w3schools.com/schema/default.asp				

XPATH					
Kurzbeschreibung:	Die XML Path Language (XPath) ist eine Abfragesprache für XML-Dokumente welche den direkten Zugriff auf Teile eines Dokumentes erlaubt. XPATH als Abfragesprache bietet neben Filterbedingungen auch so genannte Pfadausdrücke an, welche die Navigation über die Baumstruktur eines Dokumentes komfortabel unterstützen. Als Ergebnis kann ein XPath Ausdruck sowohl einfache Werte als auch eine Menge von Elementen liefert. Insbesondere definiert XPATH implizite Umwandlungsoperationen zwischen verschiedenen Ergebnistypen. Darüber hinaus bietet XPath eine breite Palette an vordefinierten Funktionen für die Verarbeitung von einfachen Datentypen als auch Knotenmengen.				
Besitzer:	W3C				
Version:	1.0	Datum:	16.10.1999	Status:	Recommendation
Version Nachfolger:	2.0	Datum:	23.01.2007	Status:	Recommendation

Zusammenhang zu anderen Standards:	XSLT, XPointer, XQuery
Gruppe:	XML Basistechnologie
Akzeptanz:	allgemein akzeptiert
Spezifikation / Quelle:	http://www.w3.org/TR/xpath20/ http://www.w3.org/TR/xpath
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/XPATH http://www.w3schools.com/xpath/default.asp

XSLT

Kurzbeschreibung:	XSL-Transformations (XSLT) ist ein Standard, welcher die Abbildung von XML-Dokumenten auf andere XML-Dokumente beschreibt. Die Abbildungsvorschrift (Stylesheet) besteht aus benutzerdefinierten Regeln, welche zum einen die Verarbeitung des Quelldokumentes und zum anderen die Erstellung des Zieldokumentes steuern. Eine Regel besteht aus einem Pattern und einem zugeordneten Template. Das Pattern filtert auf Basis eines XPATH-Ausdrucks relevante Elemente aus dem Quelldokument, welche durch das Template in das Ziel überführt werden. Ein Stylesheet ist ein XML-Dokument welches die Regeln und Templates in Form von XML-Elementen beschreibt. Die eigentliche Transformation der Dokumente erfolgt durch einen sog. XSLT-Processor. Die nachfolgende Abbildung soll diese Zusammenhänge nochmals verdeutlichen.  				
Besitzer:	W3C				
Version:	1.0	Datum:	16.11.1999	Status:	Recommendtion
Version Nachfolger:		Datum:		Status:	
Zusammenhang zu anderen Standards:	XPATH, XML, XML-Schema, XML-Namespace				
Gruppe:	XML Basistechnologie				
Akzeptanz:	allgemein akzeptiert				
Spezifikation / Quelle:	http://www.w3.org/TR/xslt				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/XSLT				

8.2 Semantische Standards

Soll die Vision der SOA von (dynamisch) austauschbaren Services wahr werden, so sind technische Standards wie WSDL, die definieren, wie ein Service beschrieben wird, notwendig aber nicht hinreichend. Um einen Service nutzen zu können, muss man die Semantik (Bedeutung) eines Services, also dessen Inhalte und insbesondere die auszutauschenden Daten verstehen. Stimmt eine Datenstruktur oder ein Begriff (z.B. eine Rechnung oder ein Auftrag) eines Geschäftspartners nicht mit der Definition im eigenen Unternehmen überein, sind Abbildungsregeln zu definieren nach denen die Daten transformiert werden. Sind die Services selbst standardisiert, können Geschäftsprozesse vor allem im B2B-Umfeld optimiert werden und der Wechsel auf einen anderen Serviceanbieter, der inhaltlich den gleichen (standardisierten) Service anbietet, wird wesentlich erleichtert.

Abhängig vom Anwendungsfall werden diese „Standards“ auf unterschiedlichen Ebenen definiert und haben damit mehr oder weniger Relevanz:

- Firmenstandards (Services werden innerhalb der Unternehmensgrenzen definiert)
- Zwischen Partnerunternehmen (Services werden zwischen zwei oder mehreren Partnerunternehmen definiert)
- Branchenstandards (Services werden innerhalb einer Branche z. B. Banken, Versicherungen definiert)
- Weltweite Standards (Services sind weltweit definiert z. B. pop3 und imap für E-Mails)

Grundsätzlich sind bei der Betrachtung von Semantik und Services zwei Aspekte relevant. Zum einen das besser Verständnis der Semantik von Services durch entsprechende Standards zur Servicedefinition, die über die rein technische Beschreibung hinaus gehen. Zum Zweiten der Einsatz von etablierten Industriestandards deren Annäherung an die technischen Standards wünschenswert ist.

Web Service Semantik

Während die technische Beschreibung mit der WSDL (siehe Kapitel „Technische Standards“) bereits etabliert ist, gibt dieses Kapitel einen Überblick über die Standards zur semantischen Definition von Web Services.

Web Service Semantics (WSDL-S)					
Kurzbeschreibung:	Der aktuelle Standard WSDL arbeitet auf der syntaktischen Ebene, damit fehlt die semantischen Ausdruckskraft, die notwendig ist, um die Anforderungen und Möglichkeiten von Web Services darzustellen. Semantik verbessert die Wiederverwendung und Entdeckung von Web Services. Sie unterstützt die Kombination von Web Services erheblich und ermöglicht die Integration von Legacy- Anwendungen als Teil der Geschäftsprozessintegration. WSDL-S definiert einen Mechanismus, um semantische Anmerkungen mit Web Services zu verbinden, die in Web Service Description Langauge (WSDL) beschrieben sind.				
Besitzer:	W3C Member Submission				
Version:	1.0	Datum:	07.11.2005	Status:	Mitglieder-einreichung
Zusammenhang zu anderen Standards:	OWL-S				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.w3.org/Submission/WSDL-S				
Weitere Links / Quellen:	Adding semantics to WSDL - White paper, http://lsdis.cs.uga.edu/library/download/wsdl-s.pdf; http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s				

Web Ontology Language for Web Services (OWL-S)

Kurzbeschreibung:	Die Web Ontology Language for Web Services (kurz OWL-S), vormals DAML-S, ist eine Spezifikation zur semantischen Auszeichnung von Web Services. OWL-S soll das Auffinden, Ausführen, Zusammensetzen und Verbinden sowie Überwachen von Web Services ermöglichen. (Quelle: Wikipedia)				
Besitzer:	DARPA Agent Markup Language Defense Advanced Research Projects Agency (DARPA)				
Version:	1.0	Datum:	11.2003	Status:	Release
Version Nachfolger:	1.2	Datum:	03.2006	Status:	Pre-Release
Zusammenhang zu anderen Standards:	WSDL-S				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.daml.org/services/				
Weitere Links / Quellen:	Wikipedia: http://de.wikipedia.org/wiki/Web_Ontology_Language_for_Web_Services				

Web Service Modeling Ontology (WSMO)

Kurzbeschreibung:	Die Web Service Modeling Ontology (WSMO) stellt ein Meta-Modell für die Web Service Modeling Language (WSML) dar. Sie bildet die konzeptuelle Grundlage dafür und definiert eine formale Sprache mit deren Hilfe es möglich ist, alle wichtigen Aspekte von Web Services semantisch zu beschreiben. Des Weiteren stellt sie eine Sprache für logische Ausdrücke und Regeln bereit. Sie dient also dazu Semantic Web Services zu erstellen und zu ihrer Verbreitung beizutragen und trägt somit einen Teil zum Gelingen des Semantic Web bei. (Quelle: Wikipedia)				
Besitzer:	wsmo.org				
Version:	1.3	Datum:	21.10.2006	Status:	Entwurf
Zusammenhang zu anderen Standards:	WSML				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.wsmo.org/				
Weitere Links / Quellen:	Wikipedia: http://de.wikipedia.org/wiki/WSMO				

Web Service Modeling Language (WSML)

Kurzbeschreibung:	Der Name Web Service Modeling Language (WSML) steht für eine Familie von Ontologiesprachen. Es handelt sich dabei um Sprachen, die primär mit dem Ziel entwickelt wurden, Ontologiesprachen für Semantic Web Services zur Verfügung zu stellen. Anders als bei OWL-Sprachvarianten benötigen die verschiedenen WSML-Varianten unterschiedlich komplexe Reasoner. Man muss bestimmen, welcher Grad an Ausdrucksmächtigkeit benötigt wird und hat dann die Wahl zwischen Varianten mit einfacherem/komplexerem bzw. besser skalierendem/weniger gut skalierendem Reasoner. (Quelle: Wikipedia)

Besitzer:	wsmo.org				
Version:	Draft16	Datum:	03.02.2005	Status:	Entwurf
Zusammenhang zu anderen Standards:	WSMO				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.wsmo.org/wsml/				
Weitere Links / Quellen:	Wikipedia: http://de.wikipedia.org/wiki/WSML				

Semantic Web Service Framework (SWSF)

Kurzbeschreibung:	SWSF bildet den Rahmen um die Standards SWSL und SWSO.				
Besitzer:	W3C Member Submission				
Version:		Datum:	09.09.2005	Status:	Mitgliedereinreichung
Zusammenhang zu anderen Standards:	SWSL, SWSO				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.w3.org/Submission/SWSF/				
Weitere Links / Quellen:					

Semantic Web Service Language (SWSL)

Kurzbeschreibung:	SWSL ist eine logikbaisierte Sprache, um formale Charakteristika von Web Service Konzepten und Beschreibungen von individuellen Services zu spezifizieren.				
Besitzer:	W3C Member Submission				
Version:		Datum:	09.09.2005	Status:	Mitgliedereinreichung
Zusammenhang zu anderen Standards:	SWSL, SWSF				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.w3.org/Submission/SWSF-SWSL/				
Weitere Links / Quellen:					

Semantic Web Service Ontology (SWSO)					
Kurzbeschreibung:	Die Semantic Web Services Ontology (SWSO) bietet ein konzeptionelles Modell mit dem ein Web Service beschrieben werden kann und eine Axiomatisierung oder formale Charakterisierung dieses Modells.				
Besitzer:	W3C Member Submission				
Version:		Datum:	09.09.2005	Status:	Mitglieder-einreichung
Zusammenhang zu anderen Standards:	SWSF, SWSO				
Gruppe:	Semantic Web Services				
Akzeptanz:					
Spezifikation / Quelle:	http://www.w3.org/Submission/SWSF-SWSO/				
Weitere Links / Quellen:					

Industiestandards

Die Bestrebung des semantischen Webs stecken zum großen Teil noch im Forschungsstadium. Während damit verbundene Standards darauf abzielen Web Services um semantische Aspekte zu erweitern, hat sich die Industrie in vielen Bereichen auf Standards geeinigt, um die Geschäftsprozesse im B2B-Bereich zu optimieren. In diesem Kapitel erfolgt eine Aufzählung etabliertet Standards. Die Liste erhebt keinen Anspruch auf Vollständigkeit und soll über die Community zu diesem Leitfaden weiter wachsen. Es wäre zu wünschen, dass diese Standards – wo noch nicht geschehen - mit denen der SOA zusammenwachsen.

XBRL eXtensible Business Reporting Language					
Kurzbeschreibung:	XBRL (eXtensible Business Reporting Language) ist eine auf XML basierende Sprache, mit der elektronische Dokumente im Bereich der Finanzberichterstattung erstellt werden. Insbesondere werden Jahresabschlüsse in dieser Sprache generiert. (Quelle: Wikipedia)				
Besitzer:	XBRL International XBRL Deutschland e.V.				
Version:	2.1	Datum:	31.12.2003	Status:	Empfehlung
HGB-Taxonomie	2.0				
Zusammenhang zu anderen Standards:					
Akzeptanz:	allgemein akzeptiert!				
Spezifikation / Quelle:	http://www.xbrl.org/Specifications/ HGB-Taxonomie: http://www.xbrl.de/index.php?option=com_content&task=view&id=56&Itemid=46				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/XBRL				

ebXML Electronic Business XML	
Kurzbeschreibung:	ebXML steht für Electronic Business XML, d. h. XML für elektronische Geschäftsprozesse. ebXML ist eine 1999 gestartete, gemeinsame Initiative von UN/CEFACT und OASIS. Ziel der Initiative ist die Entwicklung eines technischen Rahmens zur Nutzung von XML für elektronische Geschäftsprozesse sowie eine Senkung der Eintrittsbarrieren für klein- und mittelständische Unternehmen (KMU) und Entwicklungsländer. ebXML ist kein Standard an sich, sondern eine Familie verschiedener Standards von UN/CEFACT und OASIS. Zu den ebXML-Standards gehören u. a. die grundlegende technische ebXML-Architektur (ebXML Technical Architecture Specification), ein XML-Schema für Geschäftsprozesse (Business Process Specification Schema), ein Registrierdienst (Registry Services Specification) mit einem Registry Information Model ebRIM und ein Nachrichtendienst (Message Service Specification). (Quelle: Wikipedia)
Besitzer:	UN/CEFACT OASIS
Zusammenhang zu anderen Standards:	
Akzeptanz:	allgemein akzeptiert!
Spezifikation / Quelle:	http://www.ebxml.org/
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/EbXML

FinTS - Financial Transaction Services					
Kurzbeschreibung:	FinTS steht für Financial Transaction Services und ist die Weiterentwicklung des 1996 erstmals vom ZKA (Zentraler Kreditausschuss) veröffentlichten Online-Banking Standards: „Homebanking Computer Interface (HBCI)“. Damals wie heute ist das Ziel dieses Standards die Vereinheitlichung der Schnittstelle zwischen dem Bankkunden - z. B. repräsentiert durch seine Finanzverwaltungs-Software - und einem oder mehreren Kreditinstituten in identischer Weise. Ziel ist dabei die Multibankfähigkeit. (Quelle: http://www.hbci-zka.de/)				
Besitzer:					
Version:	4.0	Datum:	09.07.2004	Status:	Release
Zusammenhang zu anderen Standards:					
Akzeptanz:					
Spezifikation / Quelle:	http://www.hbci-zka.de/spec/fints_v4_0.htm				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/FinTS				

eTOM - Enhanced Telecom Operations Map

Kurzbeschreibung:	Die enhanced Telecom Operations Map (eTOM) ist ein Rahmenwerk für Geschäftsprozesse von Unternehmen der Informations- und Telekommunikationsindustrie, das vom TeleManagement Forum herausgegeben wird. Der eTOM-Ansatz beruht darauf, dass Telekom- und IT-Firmen häufig Daten austauschen müssen, um über eine Prozesskette hinweg gegenüber einem Endkunden eine Leistung zu erbringen. Um einen qualitativ möglichst guten Service erbringen zu können, müssen alle Prozessbeteiligten dieselben Vorstellungen von Qualität in Bezug auf die versprochenen Eigenschaften der Leistung haben. Eine Voraussetzung ist daher die Transparenz der Geschäftsprozesse über die beteiligten Unternehmen hinweg. Am leichtesten ist dies zu erreichen, wenn diese Unternehmen ihre Prozesse an Standards ausrichten, und solche Standards versucht das eTOM vorzugeben. Das bedeutet nicht, dass ein solches Unternehmen alle eTOM-Prozesse in ihrer Vollständigkeit umsetzen muss: das eTOM versteht sich als Angebot zum Nutzen aller Beteiligten. (Quelle: Wikipedia)				
Besitzer:	Telemanagement-Forum				
Version:	6.0	Datum:	01.01.2006	Status:	Release
Zusammenhang zu anderen Standards:	SID				
Akzeptanz:					
Spezifikation / Quelle:	http://www.tmforum.org/page31903.aspx				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/ETOM				

SID - Shared Information and Data

Kurzbeschreibung:	Das SID Modell (Shared Information and Data) wird für die Telekommunikationsindustrie entwickelt. Verschiedene große Telekommunikationsunternehmen, die Mitglieder der Organisation TeleManagement Forum sind, unterstützen über diesen Zusammenschluss die Entwicklung des generischen Informations- und Datenmodells. SID repräsentiert laut seinen Autoren eine logische Sicht auf Objekte (Entitäten) die innerhalb eines Telekommunikationsunternehmen von Interesse sind. Darüber hinaus werden auch die Beziehungen (Assoziationen) zwischen den Entitäten definiert. (Quelle: Wikipedia)				
Besitzer:	Telemanagement-Forum				
Version:	6.0	Datum:	01.01.2006	Status:	Release
Zusammenhang zu anderen Standards:	eTOM				
Akzeptanz:					
Spezifikation / Quelle:	http://www.tmforum.org/page31908.aspx				
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/Shared_Information_%26_Data_Model				

EDIFACT - Electronic Data Interchange For Administration, Commerce and Transport	
Kurzbeschreibung:	UN/EDIFACT ist die Abkürzung für United Nations Electronic Data Interchange For Administration, Commerce and Transport. EDIFACT ist ein branchenübergreifender internationaler Standard für das Format elektronischer Daten im Geschäftsverkehr. EDIFACT ist einer von mehreren internationalen EDI-Standards. Verantwortlich für den EDIFACT-Standard ist eine UN-Einrichtung namens CEFACT, die der UNECE angegliedert ist. (Quelle: Wikipedia)
Besitzer:	Economic Commission for Europe, (UN/ECE)
Zusammenhang zu anderen Standards:	
Akzeptanz:	
Spezifikation / Quelle:	http://www.unece.org/trade/untdid/directories.htm
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/EDIFACT

RosettaNet	
Kurzbeschreibung:	RosettaNet ist ein Non-Profit-Konsortium von weltweit mehr als 600 Unternehmen, vornehmlich aus den Bereichen Informationstechnologie, elektronische Bauelemente und Halbleiterfertigung, Logistik, Telekommunikation, Dienstleistung etc. Ziel ist eine Standardisierung von Datenübertragungsschnittstellen im gegenseitigen elektronischen Datenaustausch. Im Rahmen der RosettaNet-Organisation werden unter Benutzergruppen und Mitgliedern branchenübergreifende und offengelegte Kommunikations- und Ablaufverfahren zum Austausch von Geschäftsdokumenten zwischen den EDV-Systemen der Nutzer über elektronischen Datenaustausch vereinbart und standardisiert, um den Datenaustausch zwischen Lieferanten und Kunden möglichst frei von Medienbrüchen und Datenkonvertierungsproblemen und damit kostengünstiger, schneller und genauer abwickeln zu können (B2B). Dabei stehen hauptsächlich die Bereiche der Logistik und Produktion im Vordergrund, aber auch der Austausch von Produkt- und Materialdaten sowie die Abwicklung von Serviceprozessen werden berücksichtigt. (Quelle: Wikipedia)
Besitzer:	RosettaNet
Zusammenhang zu anderen Standards:	
Akzeptanz:	
Spezifikation / Quelle:	http://portal.rosettanet.org/cms/sites/RosettaNet/
Weitere Links / Quellen:	http://de.wikipedia.org/wiki/Rosettanet

Glossar

AJAX	Aynchronous JavaScript und XML	AJAX ermöglicht die asynchrone Datenübertragung zwischen Client und Server. Dadurch können lediglich einzelne Elemente von HTML-Seiten bei Bedarf dynamisch nachgeladen werden (mit JavaScript). HTTP-Requests können durchgeführt werden, ohne dass die entsprechende Seite komplett neu geladen werden muss.
BPEL	Business Process Execution Language	BPEL ist eine Sprache zur Beschreibung des Verhaltens von Geschäftsprozessen auf der Basis von WebServices.
BPM	Business Process Management	Das Geschäftsprozessmanagement beschäftigt sich mit dem Herausfinden, Gestalten, Dokumentieren, Optimieren, Überwachen und Steuern Geschäftsprozessen.
BPMN	Business Process Modeling Notation	BPMN ist eine grafische Spezifikationssprache zur visuellen Dokumentation der Struktur, der Organisation sowie des Verhaltens eines Unternehmens.
DAML	DARPA Agent Markup Language	DAML ist eine Ontologiesprache für das Semantic Web. Dabei handelt es sich um einen veralteten Standard, der durch OWL abgelöst wurde.
Dienst		Der Begriff Dienst bezeichnet ein Konzept aus dem Kontext der Geschäftsprozessmodellierung, das für eine wiederholbare Ausführung von Geschäftsaktivitäten steht. Um das Konzept Dienst auf eine DV-technische Realisierung abzubilden, werden Dienste als Entitäten betrachtet, die ähnlich wie Komponenten Funktionalität über Schnittstellen bereitstellen. Im Gegensatz zu Komponenten wird jedoch von den Abhängigkeiten der die Dienste erbringenden Entitäten abstrahiert. Wie Komponenten können Dienste externe Ressourcen verwalten. Dienste werden immer für eine dedizierte Dienstplattform entwickelt, die ein Kommunikationsprotokoll und -mechanismen definiert, auf deren Grundlage eine Kommunikation zwischen Diensten bzw. zwischen Dienstnutzern und Diensten erfolgen kann. Höherwertige Dienste können aus bestehenden Diensten komponiert werden, z. B. auf der Basis von BPEL-Spezifikationen. Dienste können in Form von Dienst-Komponenten neu entwickelt werden, es ist jedoch auch möglich, Funktionalität bestehender Anwendungen zu kapseln und als Dienst zur Verfügung zu stellen.

DTD	Document Type Definition	Die Struktur (Grammatik) von XML-Dokumenten kann über eine so genannte Document Type Definition (DTD) beschrieben werden. Eine DTD definiert den Aufbau jedes einzelnen Elements, indem die Struktur der Unterelemente definiert wird. Der Inhalt eines Elementes setzt sich aus weiteren Elementen, Text oder einer Mischform aus beidem zusammen. Ein XML-Dokument wird als „valid“ bezeichnet, wenn die Dokumentstruktur der in der DTD beschriebenen Syntax entspricht. Diese Prüfung kann automatisch durch einen Parser erfolgen.
EAI	Enterprise Application Integration	EAI ist ein Konzept zur unternehmensweiten Integration der Geschäftsfunktionen entlang der Wertschöpfungskette, die über verschiedene Applikationen auf unterschiedlichen Plattformen verteilt sind, und die im Sinne der Daten- und Geschäftsprozessintegration verbunden werden können. EAI umfasst dabei die Planung, die Methoden und die Software, um heterogene, autonome Anwendungssysteme prozessorientiert zu integrieren.
ebXML	Electronic Business XML	s. Kapitel „Technische Standards“
EDA	Event Driven Architecture	Eine Ereignisgesteuerte Architektur ist eine Softwarearchitektur, in der das Zusammenspiel der Komponenten durch Ereignisse (events) gesteuert wird. Ereignisse können sowohl von außen kommen (z. B. Benutzereingaben) als auch vom System selbst ausgelöst werden (z. B. Änderungsbenachrichtigungen). Sie stoßen eine Ereignisbehandlung (event handling) an, mit der das System auf die erfolgte Eingabe reagiert. Ereignisgesteuerte Architekturen stellen einen Spezialfall von Serviceorientierten Architekturen dar.
EDIFACT	Electronic Data Interchange For Administration, Commerce and Transport	s. Kapitel “Semantische Standards”
EPK	Ereignisgesteuerte Prozesskette	EPK dienen zur Modellierung von Geschäftsprozessen und Arbeitsabläufen einer Organisation. Durch eine Modellierungssprache werden diese grafisch dargestellt.
eTOM	enhanced Telecom Operations Map	s. Kapitel „Semantische Standards“
FinTS	Financial Transaction Services	s. Kapitel „Semantische Standards“

ITIL	IT Infrastructure Library	Die ITIL ist eine Bibliothek, welche den Standard zur Beschreibung des IT-Service-Managements darstellt. Die Inhalte des Rahmenwerks orientieren sich am Lebenszyklus des Service: Strategie (Strategy), Entwurf (Design), Betriebsüberleitung (Transition), Betrieb (Operation) und Verbesserung (Continual Improvement). In dem Regel- und Definitionswerk werden die für den Betrieb einer IT-Infrastruktur notwendigen Prozesse, Aufbauorganisation und Werkzeuge beschrieben.
JBI (JSR 208)	Java Business Integration	s. Kapitel „Technische Standards“
JCP	Java Community Process	JCP beschreibt das Verfahren zur Weiterentwicklung von Java, d.h. es wird der Prozess definiert, den ein Vorschlag zur Weiterentwicklung durchlaufen muss, bevor er tatsächlich zum offiziellen Bestandteil der Sprache wird.
Komponente		Eine Komponente bezeichnet eine Software-Entität, die ohne Änderung in Software-Systemen verwendet werden kann, die außerhalb der Kontrolle der Entwickler der Komponente liegen. Nutzer haben keinen Zugriff auf den Quellcode einer Komponente, können deren Verhalten jedoch in der von den Entwicklern der Komponente vorgesehenen Art und Weise anpassen. Komponenten können die von anderen Komponenten bereitgestellte Funktionalität nutzen oder aus anderen Komponenten komponiert werden. Komponenten haben Laufzeiteigenschaften und können externe Ressourcen verwalten. Komponenten sind immer an eine Komponentenplattform gebunden und können nur in eine für den entsprechenden Komponententyp geeignete Komponentenlaufzeitumgebung verteilt und dort betrieben werden.
LOB	Line of Business (dt. Geschäftszweig)	
Modul		Die Begriffe „Module“ und „Komponente“ werden oft austauschbar benutzt. Der Begriff „Modul“ bezeichnet ein Strukturierungselement in der Entwurfsphase von Software-Entitäten. Module dienen zur Dekomposition von Programm-Codes. Ein Modul kann in Beziehung zu anderen Modulen stehen. Da Module Elemente der Entwurfsphase sind, besitzen sie keine Laufzeiteigenschaften.
OASIS	Organization for the Advancement of Structured Information Standards	Die OASIS ist eine Non-Profit-Organisation, welche sich mit der Entwicklung und Pflege von Web Service- und e-Business-Standards beschäftigt. Eine Übersicht über Standards, an denen die OASIS teilhat, finden Sie unter http://www.oasis-open.org/specs.

OWL-S	Web Ontology Language for Web Services	s. Kapitel „Semantische Standards“
QoS	Quality of Service	QoS ist die Sammlung der Qualitätsanforderungen an einen Service.
RosettaNet		s. Kapitel „Semantische Standards“
SaaS	Software as a Service	SaaS ist ein Geschäftsmodell mit der Idee, Software als Dienstleistung anzubieten, d.h. diese nach Bedarf - insbesondere über das Web – bereitzustellen.
SAML	Security Assertion Markup Language	s. Kapitel „Technische Standards“
SCA	Service Component Architecture	SCA ist eine Spezifikation zur Vereinheitlichung des Zugriffs auf Services. Dem SOA-Gedanken folgend sind SCA Komponenten unabhängig von einer konkreten Technologie.
Schnittstelle		In der Softwaretechnik versteht man unter einer Schnittstelle die Grenzlinie, an der sich zwei unabhängige Software-Bausteine berühren und miteinander interagieren oder kommunizieren. Eine Schnittstelle ist mehr als eine Liste verfügbarer Dienste oder Operationen. Idealerweise stellt die Spezifikation einer Schnittstelle genügend Informationen bereit, um unerwartete Interaktionen zu vermeiden, die auf Grund von Annahmen auftreten können, die ein Baustein über nicht spezifizierte Eigenschaften seiner Umgebung oder anderer an einem Kommunikationsprozess beteiligter Bausteine macht.
SID	Shared Information and Data	s. Kapitel „Semantische Standards“
SLA	Service-Level-Agreement	Ein SLA ist eine Vereinbahrung zwischen Auftragsgeber und Auftragsnehmer über Leistungseigenschaften wie etwa Reaktionszeit, Umfang und Schnelligkeit von Services, sozusagen das Level der Dienstgüte (s. auch Quality of Service).
SOAD	Service-Oriented Analysis and Design	SOAD beschreibt Entwicklungsprozess und Notationen für eine SOA.
SOAP	Simple Object Access Protocol	s. Kapitel „Technische Standards“

Software-System		Ein Software-System ist eine Software-Entität, die dem Anwender eine komplexe Funktionalität zur Verfügung stellt. Ohne Beschränkung der Allgemeinheit können Software-Systeme zumindest konzeptionell auf das 4-Schichten-Modell abgebildet werden. Moderne Software-Systeme werden aus Komponenten zusammengesetzt und nutzen ggf. bereitgestellte Dienste, um ihre Funktionalität zu erbringen.
SWSF	Semantic Web Services Framework	SWSF bildet den Rahmen um die Standards SWSL und SWSO.
SWSL	Semantic Web Service Language	s. Kapitel „Semantische Standards“
SWSO	Semantic Web Services Ontology	s. Kapitel „Semantische Standards“
TCO	Total Cost of Ownership	TCO ist ein Verfahren, das Unternehmen ermöglicht, Kosten von Investitionsgütern abzuschätzen. Es wird insbesondere bei der Beschaffung und Betreuung von IT-Vermögensgegenständen angewandt. Die Idee ist, nicht nur die Anschaffungskosten, sondern auch die später anfallenden Kosten, etwa durch Wartung von Systemen, miteinzubeziehen.
UDDI	Universal Description, Discovery and Integration	s. Kapitel “Technische Standards”
UML	Unified Modelling Language	UML ist eine standardisierte Sprache zur Beschreibung von Software und anderen Systemen. Sie wurde von OMG (Object Management Group) entwickelt.
WS	Web Service	Ein Web Service ist ein Standard, der Funktionen in verteilten Umgebungen zur Verfügung stellt.
WS-Addressing		Definition eines SOAP Headers bzw. Mechanismus, um Web Services transportneutral zu adressieren.
WSDL	Web Service Description Language	s. Kapitel „Semantische Standards“
WS-I	Web Services Interoperability Organization	Die WS-I ist eine Standardisierungsorganisation von Web Services zur Gewährleistung von deren Kompatibilität in verschiedenen Implementierungen.

WS – MetadataExchange		s. Kapitel „Technische Standards“
WSMO	Web Service Modeling Ontology	s. Kapitel “Semantische Standards”
WSML	Web Service Modeling Language	s. Kapitel „Semantische Standards“
WS-Policy		WS-Policy beschreibt die Eigenschaften, Anfoderungen und allgemeinen Charakteristika eines Webdienstes
WS-Reliable Messaging		s. Kapitel „Technische Standards“
WS-SecureConversation		s. Kapitel „Technische Standards“
WS-Security		s. Kapitel „Technische Standards“
WS-Tranfer		Definiert Mechanismen zum Zugriff XML-basierter Repräsentationen von Ressourcen (Entitäten) unter der Verwendung der Web Services Infrastruktur.
W3C	World Wide Web Consortium	Beim W3C handelt es sich um eine Standardisierungsorganisation, die sich mit den Techniken rund um das World Wide Web beschäftigt.
XBRL	eXtensible Business Reporting Language	s. Kapitel „Semantische Standards“
XML	Extensible Markup Language	s. Kapitel „Technische Standards“
XSD	XML-Schema	s. Kapitel „Technische Standards“
XPATH	XML Path Language	s. Kapitel „Technische Standards“
XSLT	XSL-Transformations	s. Kapitel „Technische Standards“

Danksagung

Besonderer Dank gilt dem BITKOM-Arbeitskreis SOA-Technologies, insbesondere den Autoren des Leitfadens:

Kapitel 1 Einführung

1.1	Zielsetzung und Motivation des Leitfadens	Dr. Ulrich Kriegel, Fraunhofer-Institut für Software- und Systemtechnik (FhG ISST) Karin Sondermann, Microsoft Deutschland GmbH Stephan Ziegler, BITKOM e. V.
1.2	Organisation und Aufbau des Leitfadens	Stephan Ziegler, BITKOM e. V.
1.3	SOA: Definition und Abgrenzung	Sebastian Beckert, SAP AG Joachim Moser, GFT Technologies AG Friedrich Vollmar, IBM Deutschland GmbH Karin Sondermann, Microsoft Deutschland GmbH Stephan Ziegler, BITKOM e. V.
1.4	1Für wen ist die Serviceorientierte Architektur relevant?	Felix Ott, DATEV eG Karin Sondermann, Microsoft Deutschland GmbH Emmanuel Bloch, T-Systems Enterprise Services GmbH Friedrich Vollmar, IBM Deutschland GmbH

Kapitel 2 Nutzen / Treiber

2.1	Nutzen und Notwendigkeit	Karin Sondermann, Microsoft Deutschland GmbH Friedrich Vollmar, IBM Deutschland GmbH Felix Ott, DATEV eG
2.2	Problemstellung und Lösungsansätze	Karin Sondermann, Microsoft Deutschland GmbH
2.3	Erfolgsfaktoren	Fritz Absmaier, IBM Deutschland GmbH Friedrich Vollmar, IBM Deutschland GmbH Karin Sondermann, Microsoft Deutschland GmbH Felix Ott, DATEV eG
2.4	Potentielle Risiken und geeignete Maßnahmen	Dr. Andreas Bungert, Arcway AG Dr. Ulrich Kriegel, Fraunhofer-Institut für Software- und Systemtechnik

Kapitel 3 Vorbereitung und Planung

3.1	SOA-Readiness	Georg Ember, IBM Deutschland GmbH Uwe Zeithammer, Fujitsu Services GmbH Ralf Konrad, T-Systems Enterprise Services GmbH Harald Ließmann, Cisco Systems GmbH Andreas Herzig, Compuware GmbH Emmanuel Bloch, T-Systems Enterprise Services GmbH
3.2	SOA-Projektplanung und Vorgehen	Dr. Thomas Ernst, Hewlett-Packard GmbH Dirk Goldhan, incowia GmbH Frank Schwarz, Saxonia Systems AG Marian Kuffner, Sopera GmbH Georg Ember, IBM Deutschland GmbH
3.3	Messung des Return on Investment	Kai-Uwe Schäfer, Freier Berater
3.4	Migrationsplanung	Dr. Ulrich Kriegel, Fraunhofer-Institut für Software- und Systemtechnik Thomas Klauß, BITKOM e. V. Stephan Ziegler, BITKOM e. V.
3.5	Value Chain - Wertschöpfungsketten	Dr. Bert Klöppel, T-Systems Enterprise Services GmbH Emmanuel Bloch, T-Systems Enterprise Services GmbH
3.6	Governance-Aufbau	Marian Kuffner, Sopera GmbH Georg Ember, IBM Deutschland GmbH Uwe Zeithammer, Fujitsu Services GmbH Stephanie Hirschberger, IBM Deutschland GmbH
3.7	Strategie und Taktik	Stefan Thurow, Avanade Deutschland GmbH

Kapitel 4 Architektur und Aufbau

4.1	Mapping: Business und IT	Ralf Konrad, T-Systems Enterprise Services GmbH Fritz Absmaier, IBM Deutschland GmbH

4.2	Architektur-Entwurf	Alexander Ockl, Freier Berater Joachim Moser, GFT Technologies AG Kai-Uwe Schäfer, Freier Berater Peter Mengel, InterSystems GmbH Dr. Bert Klöppel, T-Systems Enterprise Services GmbH Harald Ließmann, Cisco Systems GmbH Wilfried Fox, Siemens IT Solutions and Services Dr. Bernd Hafenrichter, EMDS AG
4.3	Security	Harald Ließmann, Cisco Systems GmbH Friedrich Vollmar, IBM Deutschland GmbH Lars Hennig, IBM Deutschland GmbH Frank Eisenhardt, IBM Deutschland GmbH

Kapitel 5 Umsetzung und Betrieb

5.1	SOA-spezifisches Vorgehen	Uwe Zeithammer, Fujitsu Services GmbH Andreas Jung, PricewaterhouseCoopers AG Dr. Ulrich Kriegel, Fraunhofer-Institut für Software- und Systemtechnik
5.2	Governance-Umsetzung	Friedrich Vollmar, IBM Deutschland GmbH Georg Ember, IBM Deutschland GmbH Uwe Zeithammer, Fujitsu Services GmbH Stephanie Hirschberger, IBM Deutschland GmbH Frank Kertscher, IBM Deutschland GmbH
5.3	Erweiterte Nutzenmodelle	Andreas Herzig, Compuware GmbH Karin Sondermann, Microsoft Deutschland GmbH

Kapitel 6 Ausblicke

	Ausblicke	Dr. Sabine Thürmel, Siemens Enterprise Communications GmbH & Co. KG Georg Ember, IBM Deutschland GmbH Wolfgang Beinhauer, Fraunhofer Institut für Arbeitswirtschaft und Organisation Friedrich Vollmar, IBM Deutschland GmbH

Kapitel 7 Soft und Security

7.1	Einleitung	Dr. Federico Crazzolara, Datev e. G. Friedrich Vollmar, IBM Deutschland GmbH Linda Strick, Fraunhofer FOKUS

7.2	Geschäftsprozesse und deren Security-Anforderungen	Dr. Federico Crazzolara, Datev e. G.
7.3	Policy Management	Günter Waller, IBM Deutschland GmbH Detlef Sturm, Beta Systems Software AG
7.4	Risikomanagement in einer SOA-Welt	André Hohner, Siemens IT Solutions and Services GmbH & Co. OHG
7.5	Security-Implementierungen	Achim Reckeweg, Sun Microsystems GmbH Dr. Thomas Störtkuhl, Secaron AG
7.6	Wechselwirkung zwischen Performanz, Verfügbarkeit und Sicherheit in einer SOA	Rüdiger Kern, IBM Deutschland GmbH Eva Saar, T-Systems Enterprise Services GmbH André Hohner, Siemens IT Solutions and Services GmbH & Co. OHG

Kapitel 8 Standards

8.1	Technische Standards	Alexander Ockl, Freier Berater Dr. Ulrich Kriegel, Fraunhofer-Institut für Software- und Systemtechnik Wolfgang Beinhauer, Fraunhofer Institut für Arbeitswirtschaft und Organisation Marian Kuffner, Sopera GmbH Stephan Ziegler, BITKOM e. V.
8.2	Semantische Standards	Dr. Bernd Hafenrichter, EMDS AG Wolfgang Beinhauer, Fraunhofer Institut für Arbeitswirtschaft und Organisation Marian Kuffner, Sopera GmbH Stephan Ziegler, BITKOM e. V. Felix Ott, DATEV eG Alexander Ockl, Freier Berater Mario Wendt, Microsoft Deutschland GmbH

Der Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. vertritt mehr als 1.200 Unternehmen, davon 900 Direktmitglieder mit etwa 135 Milliarden Euro Umsatz und 700.000 Beschäftigten. Hierzu zählen Anbieter von Software, IT-Services und Telekommunikationsdiensten, Hersteller von Hardware und Consumer Electronics sowie Unternehmen der digitalen Medien. Der BITKOM setzt sich insbesondere für bessere ordnungspolitische Rahmenbedingungen, eine Modernisierung des Bildungssystems und eine innovationsorientierte Wirtschaftspolitik ein.

BITKOM

Bundesverband Informationswirtschaft,
Telekommunikation und neue Medien e.V.

Albrechtstraße 10 A
10117 Berlin-Mitte
Tel.: 030.27576-0
Fax: 030.27576-400
bitkom@bitkom.org
www.bitkom.org