

Whitepaper

Was bringt SOA dem gehobenen Mittelstand?

Ein pragmatischer, praxisorientierter Ansatz

Autor: Lars Brömstrup, Dr. Michael Bark, Wolfgang Wulf

Datum: 9. Oktober 2006, Version 1.0

© evodion Information Technologies GmbH

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

1	SOA – auch eine Chance für den gehobenen Mittelstand?	3
2	Der pragmatische, praktisch orientierte SOA-Projektansatz	5
2.1	evodion-SOA: unser Vorschlag für ein Vorgehenskonzept für SOA-Projekte:	6
2.2	Beispiele für Services	7
2.3	Zusammenfassung	9
3	Case Study SPIEGEL Verlag	11

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

1 SOA – auch eine Chance für den gehobenen Mittelstand?

SOA ist in aller Munde. Einige Großunternehmen und Konzerne haben bereits (Pilot)-Projekte in diesem Umfeld gestartet, um die Anwendbarkeit in ihrem Unternehmen zu erproben und zu beweisen. In den einschlägigen Medien wird auch in einem großen Stile über SOA berichtet. Der Tenor dieser Berichterstattung ist breit gefächert und reicht von „SOA lohnt sich nur für die wenigsten Firmen“ bis hin zu „Nur mit SOA bleiben Firmen wettbewerbsfähig“. Das führt zu einer Verunsicherung am Markt, obwohl bei vielen Unternehmen die Anforderungen vorhanden sind, die die Implementierung von SOA rechtfertigen:

„Heute die Integration der neuen Tochtergesellschaft, morgen die Aufforderung der Geschäftspartner Geschäftsdaten elektronisch auszutauschen, übermorgen eine Lösung für die globale Vertriebsmannschaft, und nächste Woche die Auslagerung von Verwaltungsaufgaben“. Die geschäftlichen Herausforderungen an die Informationstechnologie eines Unternehmens sind vielfältig und verändern sich rasch. In der Vergangenheit eingeführte monolithische IT-Systeme sind jedoch diesen Herausforderungen oft nicht gewachsen. Das gilt insbesondere für die über lange Zeiträume entstandenen Lösungen zur Planung und Abwicklung des Geschäftes (z.B. ERP-/SCM-/CRM-Lösungen). Ein Austausch dieser betriebswirtschaftlichen Kernsysteme ist kurzfristig nicht realisierbar.

Gartner sieht die Situation so wie in der nachfolgenden Skizze beschrieben: Eine große Anzahl von Anwendungen kommuniziert über eine noch größere Anzahl von Schnittstellen miteinander.

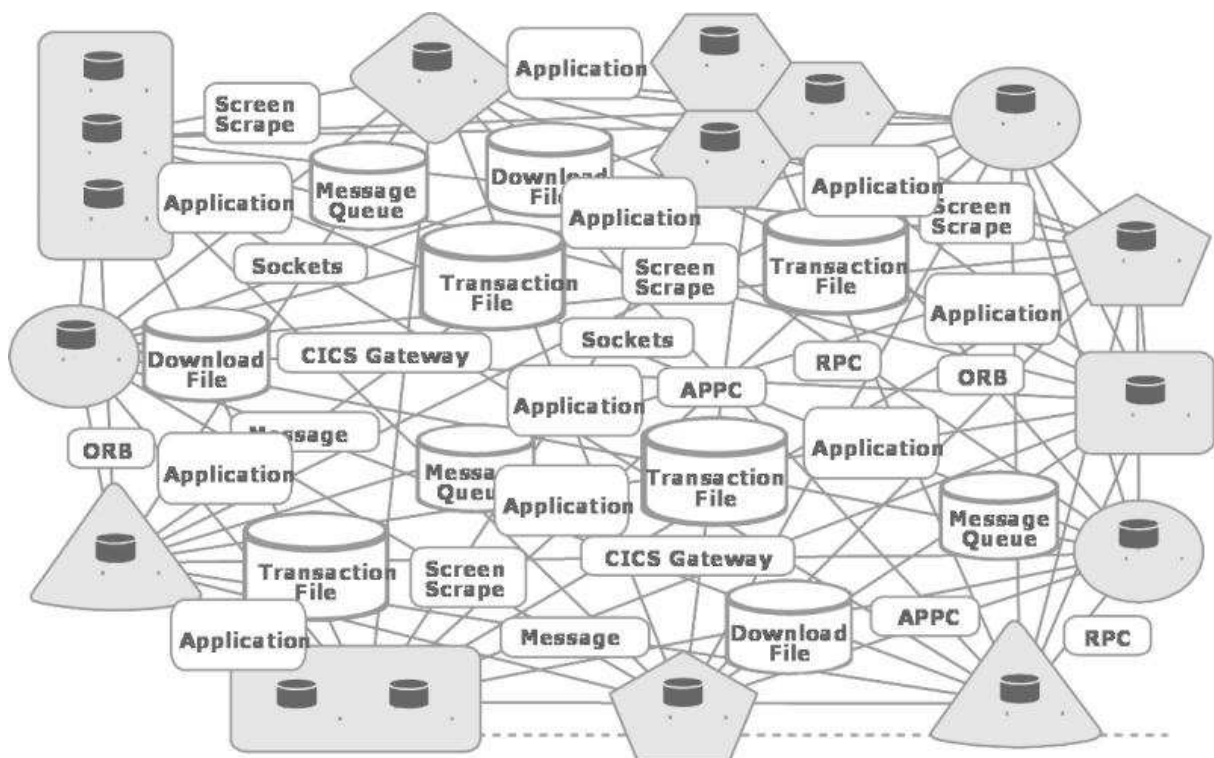


Bild: „Der gordische Knoten“, Quelle: Gartner

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

Die Komplexität ist groß, die Abhängigkeiten voneinander sind groß und die Auswirkungen von Änderungen auf andere Systeme sind nur schwierig abschätzbar.

Aber die Erwartungen der Kunden, der intern Verantwortlichen und der Partner an die Flexibilität IT-unterstützter Geschäftsprozesse sind nicht wegzureden. Was ist in dieser Situation zu tun?

In dieser Situation kann SOA helfen. Was ist nun aber SOA? SOA ist ein Architekturprinzip, eine Philosophie, aber keine Technik und wird nicht durch ein einzelnes Produkt implementiert.

Insofern ist jedes SOA-Projekt bei jedem Unternehmen unterschiedlich, muss auf die Anforderungen des Unternehmens eingehen und muss individuell konzipiert, geplant und umgesetzt werden.

In diesem Whitepaper wird ein pragmatischer Ansatz vorgestellt, der eine SOA als Architektur darstellt, die langfristig in mehreren Schritten individuell auf die Anforderungen zugeschnitten umgesetzt werden muss.

Wir haben unseren hier propagierten pragmatischen Weg mit einem Unternehmen bereits erfolgreich durchgeführt. Dieses Projektbeispiel beschreiben wir am Ende dieses Berichtes.

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

2 Der pragmatische, praktisch orientierte SOA-Projektansatz

Ein wesentlicher Aspekt von SOA ist der Service. Ein Service beschreibt eine wohl definierte, in sich abgeschlossene fachliche Funktion. Fachlich muss deshalb betont werden, weil der konventionelle Ansatz der Anwendungsarchitektur einen mehr technischen Schwerpunkt setzt, während SOA prozessorientiert die fachlichen Aspekte der Geschäftsprozesse in den Vordergrund stellt.

Services sind Einheiten der IT, die einen in sich geschlossenen Verarbeitungsschritt eines Geschäftsprozesses ausführen, zukünftig auch für weitere oder einen geänderten Geschäftsprozess genutzt werden können und sich über standardisierte beschriebene Schnittstellen aufrufen lassen. Services sind nicht dasselbe wie Objekte und können - müssen aber nicht unbedingt - durch Web-Service-Technologien realisiert werden.

SOA ist eine IT Architektur, die aus Services als atomaren Einheiten zusammengesetzt ist (anstatt aus Applikationen) und die Services auf Basis offener Standards lose koppelt. Das heißt SOA ist ein Architekturstil und kein Produkt.

Services bieten die Möglichkeit, Softwarekomponenten lose gekoppelt miteinander interagieren zu lassen. Der größte Nutzen dieser losen Kopplung ist die gewonnene Agilität, das heißt die Möglichkeit, durch Komposition dieser Services geänderte oder neue Geschäftsprozesse in kurzer Zeit abbilden zu können. Das setzt jedoch Erfahrung und Augenmaß bei der Konzeption der Granularität der Services voraus. Die Agilität stellt sich auch ein, wenn bereits vorhandene, starre Integrationen mit Hilfe einer Service Architektur aufgebrochen werden können. Vor allem verhilft die gewonnene Flexibilität zu neuen Konzepten, mit denen Kosten eingespart aber auch das Geschäft gesteigert werden können.

SOA ist eine Idee, die durch Technik unterstützt wird. Vereinfacht gesagt bildet SOA ein breit angelegtes Rahmenwerk, worin sich Softwareservices erstellen, verwalten und kombinieren lassen. Das große Ziel ist eine an Geschäftsprozessen ausgerichtete IT-Infrastruktur, die schnell auf veränderte Anforderungen reagiert, Services in verschiedenen Kontexten einsetzbar macht, somit Redundanzen abbauen hilft und Wiederverwendung praktiziert. SOA ist weder ein normierter Begriff noch ist SOA Selbstzweck, sondern SOA macht nur Sinn, wenn konkrete Anforderungen mit Hilfe von SOA besser umgesetzt werden können. SOA – und so sieht das einhellig die Fachwelt – ist eine Architektur und somit technologieneutral. Sie können mit dot.net genauso agieren wie mit CORBA, J2EE, Webservices und natürlich mit HOST-Systemen. SOA ist ein Architekturprinzip. Es beruht auf ausgereiften Konzepten und allgemein akzeptierten Standards.

Für die Einführung von SOA gibt es zwei grundsätzlich unterschiedliche Vorgehensweisen:

1. top-down Ansatz

Beim top-down Ansatz wird das gesamte Unternehmen und damit auch die IT auf SOA umgestellt. Es wird mit einer Geschäftsprozessanalyse begonnen. Die definierten Geschäftsprozesse werden danach auf Basis einer SOA implementiert.

Aus unserer Sicht hat dieses Vorgehen hohe Risiken durch mögliche kulturelle Verwerfungen, unzureichende organisatorische Maßnahmen, Überforderung der Mitarbeiter, Dimension und Laufzeit des Projektes sowie eine fehlende Lernphase und empfiehlt sich nur bei einer Situation auf der „grünen Wiese“.

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

2. bottom-up Ansatz

Der bottom-up Ansatz reduziert die oben genannten Risiken. SOA wird punktuell durch eine zusätzliche Schicht mittels Web-Service-Schnittstelle für eine bestehende Funktionalität eingesetzt. Damit steht diese Funktionalität als Dienst zur Verfügung. Die Vorteile sind ein schneller Erfolg, eine schnelle Amortisation, eine effektive Lernphase sowie ein paralleler Aufbau der Organisation.

Aber das bottom up Vorgehen realisiert nicht sofort eine Architektur im Sinne von SOA nach der „reinen Lehre“.

2.1 evodion-SOA: unser Vorschlag für ein Vorgehenskonzept für SOA-Projekte:

Wir empfehlen als Start einen bottom-up Ansatz, einen taktischen Start mit einzelnen Services:

1. Projektdefinition

Starten Sie Bottom Up. Finden Sie einen ersten Einstieg in eine SOA, ein erstes Projekt, d.h. einen Service. Das Projekt (die Implementierung des Service) muss einen praktischen Nutzen haben und außerdem die neue Architektur implementieren. Starten Sie mit einem kleinen und überschaubaren Service, der mehrfach genutzt werden soll.

2. Wählen Sie eine Technologie

Nutzen Sie Vorhandenes (Entwicklungsumgebung, etc.). Beginnen Sie mit einer SOA ohne Orchestrierung und ohne Enterprise Service Bus (ESB). Starten Sie mit einer Umsetzung von WebServices und setzen Sie bei Bedarf einen Application Servers ein. Investieren Sie lieber in gute Architekten als in neue Produkte. Wenn Sie neue Produkte kaufen, vertrauen Sie einem Anbieter, den Sie schon im Haus haben. Lassen Sie sich nicht von der Produktvielfalt verunsichern. Lassen Sie sich nicht durch äußere Einflüsse verunsichern, fokussieren Sie sich auf Ihre zu lösenden Anforderungen und verfolgen Sie damit einen eigenen SOA-Ansatz.

3. Durchführen des Projektes mit „klassischem“ Projektmanagement

Planen Sie das konkrete Projekt, vermeiden Sie weit über das Projekt hinaus zu planen, damit Sie sich auf das eine Projekt fokussieren können. Planen Sie nicht die unternehmensweite SOA, denn das ist in der Regel zu komplex. Gehen Sie pragmatisch vor, damit Sie möglichst schnell Projekterfolge erzielen. Überlegen Sie inwieweit Sie den/die Service(s) evtl. als Webservice implementieren, denn diese Technologie ist bewährt und meist schon im Hause vorhanden. Sammeln Sie die gemachten Erfahrungen und werten Sie diese aus.

4. Planung und Durchführung weiterer Projekte

Identifizieren Sie weitere Funktionen, weitere Services. Definieren Sie, wie die Services geschnitten werden müssen, damit diese mehrfach verwendbar sind/werden. Planen Sie weitere SOA-Projekte und führen Sie diese wie oben beschrieben durch. Lassen Sie die Erfahrungen des ersten Projektes einfließen. Setzen Sie weitere Services um. Nutzen Sie die Services.

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

Achten Sie beim Zuschnitt der Services auf die richtige Granularität:

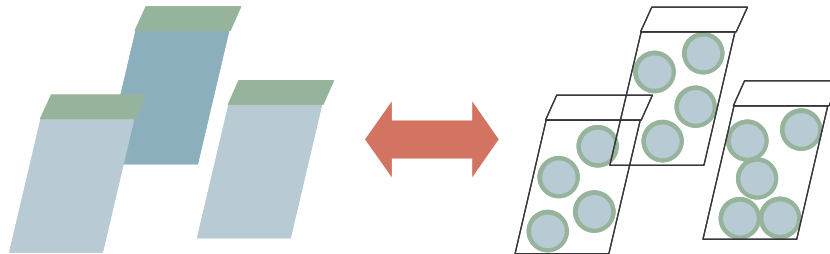


Bild 2: „Zuschnitt von Services: EAI vs. Service Zoo“

Für den konkreten Zuschnitt gibt es kein Patentrezept. Falls Sie zu groß schneiden, erhalten Sie eine Architektur, die wie EAI aussieht (linke Seite des Bildes), wenn Sie zu klein schneiden, erhalten Sie einen Service-Zoo (rechte Seite des Bildes), der schwieriger zu verwalten ist. Wichtig ist, dass Sie sich beim Zuschnitt auf die Fachlichkeit stützen und den fachlichen Zuschnitt adaptieren.

5. Ausbau der Architektur

Klären Sie beim weiteren Ausbau der Architektur, ob eine Workflow-Engine auf Basis BPEL zur Orchestrierung/Choreografie der einzelnen Services erforderlich wird, ob Sie einen Enterprise Service Bus (ESB) als Middleware bzw. Integrationsschicht zur Realisierung der Kommunikationsinfrastruktur einsetzen müssen und ob Sie bei einer zunehmenden Anzahl von Services diese in einer Registry auf Basis von UDDI verwalten. Verzichten Sie im Zweifel auf diese Bestandteile und setzen Sie eventuell vorhandene Anforderungen anders um.

Was ist an diesem Vorgehen nun pragmatisch? Der bottom-up Ansatz überwiegt, also müssen Sie keine unternehmensweite Prozessanalyse und Modellierung durchführen. Sie haben nicht (schon wieder) jahrelang Unternehmensberater im Haus. Es sind nur minimale Infrastrukturkomponenten erforderlich, da Sie keinen ESB, keine Registry (UDDI) und keine Workflowengine wie BPEL benötigen. Es gibt keine bis wenige Einflüsse auf die Organisation. Sie setzen „Learning by doing“ um und eventuell gemachte Fehler tun nicht wirklich weh. Trotzdem entwickeln Sie kein Wegwerfprodukt und haben einen hohen Nutzeneffekt.

2.2 Beispiele für Services

In der modernen Automobil-Produktion wird oft auf das sogenannte Plattformkonzept gesetzt. Das heißt, alle Autos einer Serie (oder mehrerer Serien) basieren auf der gleichen Plattform.

Wenn man dieses Beispiel jetzt auf die IT überträgt, dann ist es das Ziel der SOA eine ähnliche Standardisierung der IT zu ermöglichen. Angewandt auf das Auto-Beispiel, sind die einzelnen Komponenten eines Autos die Services in der IT-Landschaft.

Betrachten wir den Service „Motor“ näher. Dieser kann in verschiedenen konkreten Ausprägungen existieren, z.B. 1.6, 1.8 oder 2.0 Liter. Der Service „Motor“ besitzt eine konkrete Beschreibung seines Zweckes (Antrieb des Fahrzeugs). Er ist definiert, durch einen konkreten Input (Treibstoff) und einen konkreten Output (Leistung). Die Schnittstellen sind so beschrieben, dass sie mit den anderen Services des Autos (z.B. Getriebe) in Verbindung gebracht werden können. Die Interna des Motors sind sicherlich

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

für den Konstrukteur (also in der IT den Entwickler) interessant, für die fachliche Beschreibung des Services spielen diese aber keine Rolle. Ein so beschriebener Service „Motor“ ist jetzt für alle beteiligten Seiten verständlich und kann anschließend in die jeweiligen Richtungen konkretisiert werden (z.B. die Beschreibung der Interna des Motors durch den Konstrukteur).

Aber wie können umzusetzende Services konkret aussehen? Wie kann ein Szenario für ein erstes Projekt aussehen? Die folgenden Beispiele sollen dazu dienen den Begriff Service anhand von Beispielen zu verdeutlichen.

2.2.1 Adressdatenverwaltung

Häufig sind Adressen von Geschäftspartnern, Klienten und Kunden in unterschiedlichen Systemen gespeichert. Diese Systeme können abhängen von dem verkauften Produkt (zum Beispiel werden die Adressdaten bei einer Versicherung häufig jeweils beim Versicherungsvertrag gespeichert und sind damit redundant verfügbar) oder auch der verwendeten Systeme. So werden Adressdaten im Auftragsystem oder CRM-System gespeichert und dann auch noch einmal in der Buchhaltung.

Das führt zu Problemen bei Adressänderungen, bei der Dublettensuche /-identifikation von Geschäftspartnern, Kunden und Klienten.

Ein Pilotprojekt wäre somit die Ausgliederung der Adressverwaltung in einen Service, der allen Systemen zur Verfügung gestellt wird. Dabei kann eines der heutigen vorhandenen Systeme zukünftig den Service für alle anderen Systeme bereitstellen. Es besteht daher überhaupt keine Notwendigkeit den Service neu zu entwickeln, sondern auch hier wird das Paradigma genutzt, Vorhandenes zu nutzen und anderen nutzbar zu machen.

2.2.2 Mehrfachnutzung von Inhalten

Medienunternehmen verfolgen das Ziel, ihren erstellten Content mehrfach auf unterschiedlichen Wegen in unterschiedlichen Medien nutzen zu können und damit mehrfach zu vermarkten. Das heißt ein möglicher Service ist ein Content-Verwaltungs- und Bereitstellungsservice, der von mehreren Medien (Printmedien, Online-Medien, externen Nutzern,...) genutzt werden kann. Dieses kann dadurch geschehen, dass das Kernsystem der Contentverwaltung von der Clientanwendung getrennt wird und der Service über eine definierte Schnittstelle und damit dann mehreren „Nutzern“ bereitgestellt werden kann.

2.2.3 Mehrwertsteuerberechnung

Die Berechnung der Mehrwertsteuer für die Faktura und evt. auch die Berechnung der Vorsteuer für Warenlieferungen ist möglicherweise über mehrere Systeme verteilt. Bei Änderungen der Steuersätze müssen diese in allen diesen Stellen geändert werden. Ein möglicher Service stellt somit die Berechnung der Mehrwertsteuer und Vorsteuer in einem zentralen Service dar.

2.2.4 Integration von Kunden-/Mandantendaten, Kopplung interner Systeme

Häufig sind die Daten über einen Kunden oder allgemein eines Geschäftspartners über mehrere interne Systeme verteilt. Das macht die Ermittlung des Gesamt-Engagements eines Kunden (Gesamt-Auftragssumme, Geschäftsvolumen) sehr schwierig. Die Gesamtsicht auf diese Daten, wie zum Beispiel

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

dem Gesamt-Saldo eines Kunden ist aber wichtig für die Genehmigung weiterer Aufträge. Die Daten aus der Finanzbuchhaltung müssen dafür in der Auftragsverwaltung sichtbar gemacht werden.

Um dieses zu ermöglichen kann ein zentraler Service geschaffen werden, der die relevanten Daten aus den Systemen sammelt und anderen Systemen zur Verfügung stellt.

Ein ähnliches Beispiel ist die externe Bonitätsprüfung vorhandener und neuer Geschäftspartner. Auch dieser Service, der evtl. in unterschiedlichen Systemen zur Verfügung stehen muss, kann durch einen zentralen Dienst zur Verfügung gestellt werden.

Relevant kann die Integration der unterschiedlichen kundenrelevanten Systeme über eine SOA auch für kundenübergreifende Geschäftsvorfälle wie „Bestellung bearbeiten“, „Auftragsliste erstellen“, „Kundenliste erstellen“, „Eingang Reklamation“ sein.

Auch für die Vertriebsunterstützung in Bezug auf Kundenklassifizierung ist eine Kopplung von CRM und Abwicklung über SOA mit Webservices und nicht über Schnittstellen ein sinnvoller Ansatz.

2.2.5 Anbindung einer (neuen) Tochterfirma

Sowohl die Integration einer bestehenden als auch einer neuen (Tochter)-Firma stellt die IT vor eine große Herausforderung. Häufig wird der Weg beschritten, die Systeme der Tochterfirma abzulösen und die vorhandenen Systeme des Mutterunternehmens einzuführen. Natürlich mag es dafür gute Gründe geben, aber es gibt auch Gründe die Integration über eine SOA zu realisieren und die vorhandenen Systeme (evtl. nur für eine Übergangszeit) beizubehalten und über Webservices zu koppeln. Dabei müssen dann relevante Schnittstellen definiert werden und in mehreren Services implementiert werden. Relevant kann dieses für alle Systeme von der Auftragverwaltung über die Buchhaltung bis hin zur Personalverwaltung sein.

2.3 Zusammenfassung

Eine heterogene IT-Landschaft ist der Normalfall. Das lehrt die Realität. Und die Idee einer zentralen Plattform, verbunden mit Harmonisierungsgewinnen, ist verlockend. Entscheidender als die eine einzige zentrale Plattform aber ist die Einhaltung relevanter Standards. So können auch verschiedene Plattformen miteinander kommunizieren.

Selten finden IT-Projekte und damit auch SOA-Projekte auf der grünen Wiese statt, nur dort wäre ein „Big Bang“ möglich. In der Regel ist bei SOA-Projekten immer die Berücksichtigung von Vorhandenem notwendig, die Einbindung und Berücksichtigung und Integration von vorhandenen Systemen.

Starten Sie jetzt mit den bekannten Anforderungen, bauen Sie Flexibilität ein und realisieren Sie weitere Anforderungen später, damit Sie einen schnellen Erfolg erzielen.

Bei einer evolutionären Einführung von SOA beginnt man mit einfachen Szenarien, die einen dedizierten Enterprise Service Bus (ESB) noch nicht unbedingt erfordern.

Wie in diesem Dokument dargestellt ist eine SOA dadurch geprägt, dass fachliche Funktionalität, die heute teilweise redundant in Anwendungen implementiert ist, in Services ausgelagert und von mehreren Anwendungen innerhalb einer Anwendungslandschaft gemeinsam genutzt werden kann.

Der Weg hin zu einer SOA basierten Anwendungslandschaft ist lang und muss evolutionär vollzogen werden. In den ersten Phasen der Einführung von SOA in einem Unternehmen gilt es, notwendige

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

Infrastrukturen aufzubauen, Erfahrungen zu sammeln, Wissen aufzubauen und gleichzeitig alle Beteiligten in den Ablauf sich wandelnder Prozesse und Verfahren einzubeziehen.

Die wichtigste Eigenschaft von SOA ist in diesem Zusammenhang, dass SOA in kleinen Schritten eingeführt werden kann – man sollte sogar sagen muss. Starten Sie so groß wie nötig und so klein wie möglich. Führen Sie SOA in überschaubaren Schritten ein, fokussieren Sie sich auf Ihre Anforderungen. Denn bei einem pragmatischen Ansatz im Gegensatz zu einem „intergalaktisch“ akademischen Vorgehen überwiegen die Chancen und Potenziale. Sie erreichen Ihr Zielszenario mit weniger Risiken.

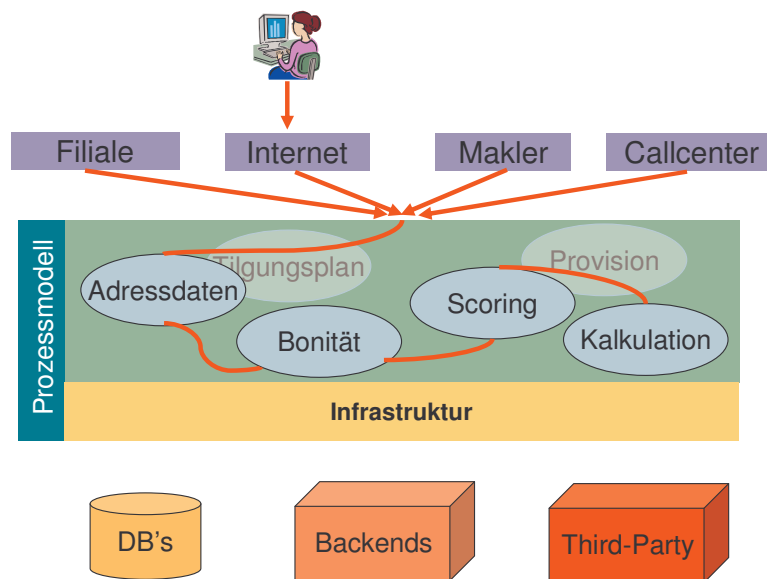


Bild 3: „Ein mögliches SOA-Zielszenario“

Was bringt SOA dem gehobenen Mittelstand? Ein pragmatischer, praxisorientierter Ansatz

3 Case Study SPIEGEL Verlag

Als moderner und innovativer Verlag hat der SPIEGEL eine Spitzenposition mit seinen Angeboten im Internet erreicht. Über besondere Schwerpunkte beim SPIEGEL im Hinblick auf Service-orientierte Architekturen berichten wir in diesem Abschnitt:

2005 hat der Spiegel eine Kooperation mit einem anderen Verlag begonnen, um mit technologischer Unterstützung der evodion ein gemeinsames Online-Archiv auf Basis des viel zitierten und sehr bekannten Spiegel Archivs zu etablieren. Schon während der Arbeiten für diese Kooperation begann der Umbau des Archivsystems Digas zu einer Service-orientierten Anwendung. Hierfür gab es vielfältige Gründe:

- Digas war eine hochkomplexe monolithische Anwendung, bei der Änderungen an einer Stelle oft weitere Änderungen an anderen Stellen nach sich zogen. Besonders betrafen dies Änderungen an der Clientschicht.
- Es gab die Anforderung, sowohl bestehende als auch neu zu entwickelnde Anwendungen um Recherchefunktionalität zu erweitern. Ohne SOA hätte dies zu redundanten, parallel zu pflegenden Recherche Komponenten geführt.
- Die Anbindung der Recherchefunktionen an andere Anwendungen soll „lose gekoppelt“ erfolgen, damit keine gegenseitige Beeinflussung der unterschiedlichen Anwendungen in den verschiedenen (DIGAS nutzenden) Verlagshäusern untereinander, aber auch mit Digas besteht.
- Die Weiterentwicklung von Digas soll unabhängig von anderen Anwendungen möglich sein.
- Das Archivsystem Digas hat einen erheblich längeren Lebenszyklus als die darauf aufbauenden Clientanwendungen.

Auch andere Anwendungen beim SPIEGEL wurden bzw. werden inzwischen auf Service-orientierte Architekturen umgestellt:

- BackOffice:
Das Back-Office ist u.a. die Basis für den gesamten Payed Content und die Abonnentenverwaltung sowie das Shopsystem. Darüber hinaus wird die Nutzung dieses Services von externen Partnern möglich (z.B. Libri-Shop). Auch eigene Anwendungen werden so vereinfacht.
- Spiegel-Online:
Bisher waren die Auslieferung der Daten sowie das Layout und die Präsentation sehr stark miteinander vernetzt, so dass Änderungen am Layout der Site sehr aufwendig waren. Durch die Einführung einer Service-orientierten Architektur können die Komponenten getrennt werden und unabhängig voneinander weiter entwickelt werden. Auch andere Ausgabegeräte wie z.B. mobile Endgeräte können jetzt einfacher unterstützt werden.

Der SPIEGEL hat sich mit Einführung einer SOA Freiraum und Unabhängigkeit für die Zukunft seiner Software geschaffen. Die Nutzungsdauer einzelner Services ist nicht mehr an den Fortschritt einzelner Dienste gebunden. Neue Services können eingefügt, nicht mehr benötigte Komponenten können entfernt und nicht mehr aktuelle Services können angepasst werden, ohne dass das ganze System aus dem Betrieb genommen werden muss.