



Hot Spots der Software-Entwicklung

Service-orientierte Architekturen – Anspruch und Wirklichkeit

21. Juli 2005

Technische Universität München
Institut für Informatik
Software & Systems Engineering
Prof. Dr. Dr. h.c. Manfred Broy
&
VSEK

Report: VSEK/035/D
Version: 1.0
Klassifikation: öffentlich

München, 22.07.2005

Marco Kuhrmann
Tilman Seifert
Gerd Beneken
Dr. Markus Pizka

Inhaltsverzeichnis

1	Einleitung	1
2	Motivation und Einleitung	5
2.1	Komplexität von IT- Systemen und Landschaften	6
2.2	Vernetzung als Komplexitätstreiber	7
2.3	Unternehmensdynamik als Komplexitätstreiber	7
2.4	Was wollen SOA's?	8
2.5	Der Workshop	9
2.6	Die Essenz	11
3	Service-orientierte Architekturen – Strategien und Visionen	13
4	Web Services – Ideale Plattform oder Mittel zum Zweck?	29
5	SOA – Von der Vision zur Realität	45
6	SOA – Zwischen Anspruch und Wirklichkeit	51
7	Wann liefert eine Serviceorientierte Architektur echten Nutzen?	55
8	Wozu SOA?	63
9	Modellbasierte dienstorientierte Software-Entwicklung	71
10	SOA – aber wie?	81
11	Sicherheit und Web Servicearchitekturen	95
12	Konzeption einer SOA Veranstalter-Plattform	107

1 Einleitung

In Unternehmen und Behörden wird eine Vielzahl unterschiedlicher IT-Systeme betrieben. Diese sind zum Teil über Jahrzehnte hin gewachsen und sich ständig ändernden Bedürfnissen angepasst worden. Die daraus resultierende wachsende Komplexität der Systeme macht deren Weiterentwicklung zu einer immer größeren Herausforderung. Eine Ablösung der Systeme kommt oft nicht in Frage, da die Kosten und Risiken zumeist nur schlecht abschätzbar sind. *Service-orientierte Architekturen* (SOA) versprechen hier eine Lösung, die es erlaubt, gewachsene Systeme zu modernisieren und sie in neu wachsende Infrastrukturen zu integrieren. Hierbei kommen verstärkt standardisierte Technologien zu Einsatz, wie http, SOAP oder auch Web Services.

Service-orientierte Architekturen – ein Begriff, der seit dem Web Services Hype der letzten Jahre immer wieder aufgegriffen wird; meistens in Verbindung mit *Enterprise Application Integration* (EAI) und *Enterprise Service Bus* (ESB). Es erscheint nahezu so, als handle es sich um eine Art allgemeingültigen Begriff, der sowohl von Softwaretechnikern als auch von Hardware- und Prozessspezialisten gebraucht wird. Viele IT-Bereiche können sich mit Services und Service-orientierten Architekturen identifizieren. Jedoch wirft dies die Frage auf: *Unterhalten wir uns alle über das Gleiche?*

Diese Frage müsste intuitiv mit Nein beantwortet werden, da in der Regel jede Sparte der IT-Landschaft eigene Probleme hat (z. B. Heterogenität, Vernetzung, schneller Wandel, Kostendruck etc.) und auch eigene Ziele verfolgt (z. B. Effizienzsteigerung, Time-to-market), die nicht immer mit denen anderer Abteilungen im Einklang stehen müssen. Es hat also den Anschein, dass Service-orientierte Architekturen hier eine Art Vereinheitlichung bringen, mit dem Anspruch, dass übergeordnete Instanzen eine einheitliche Sichtweise auf IT-Landschaften erhalten.

In der Wirklichkeit sieht das meistens anders aus, da hier Themen wie Wartbarkeit oder der Betrieb eines IT-Systems allgemein einen wesentlich höheren Stellenwert einnehmen – anerkannt ist hier zum Beispiel die *80/20* Regel für Betrieb und Entwicklung. Viele Aussagen verschiedener Hersteller und Anbieter zu Service-orientierten Architekturen zielen jedoch auf die Neuentwicklung (sei es total oder partiell) oder die Ablösung etablierter Systeme.

Diesem Unterschied zwischen Anspruch und Wirklichkeit war der Workshop „Service-orientierte Architekturen – Anspruch und Wirklichkeit“ gewidmet, in dem Referenten aus Industrie und Forschung sich zum Thema SOA positionierten, Erfahrungen aus Praxisprojekten austauschten und kritisch hinterfragten. Dieser Tagungsband fasst die Inhalte, Themen und Vorträge des Workshops zusammen.

Programm

Einführung und Begrüßung		
13:00 – 13:10	Begrüßung durch Prof. M. Broy, Lehrstuhl für Software & Systems Engineering, TU München	
13:10 – 13:20	Marco Kuhrmann (TUM): Einführung zum Workshop	
Block 1 Visionen, Technologien und Erfahrungen		
13:20 – 14:00	Klaus Rohe (Microsoft): „Service-orientierte Architekturen – Strategien und Visionen“	Seite 13
14:00 – 14:40	Dr. Harald Haller (sd&m AG): „Web Services – Ideale Plattform oder Mittel zum Zweck?“ <i>Kaffeepause</i>	Seite 29
Block 2 Kurzbeiträge – 1		
15:05 – 15:50	Gino Popescu (HVB-Systems): „SOA – Von der Vision zur Realität“	Seite 45
	Nicolai Josuttis (IT Communication): „SOA – Zwischen Anspruch und Wirklichkeit“	Seite 51
	Dr. Jan-Peter Richter (sd&m AG): „Wann liefert SOA einen echten Nutzen?“ <i>Kaffeepause</i>	Seite 55
Block 3 Grundlagen		
16:15 – 16:55	Dr. Markus Pizka (TUM): „Wozu SOA?“	Seite 63
16:55 – 17:35	Johannes Grünbauer (TUM): „Modellbasierte dienstorientierte Software-Entwicklung“ <i>Kaffeepause</i>	Seite 71
Block 4 Kurzbeiträge – 2		
17:50 – 18:30	Inge Hanschke (iteratec): „SOA – aber wie?“	Seite 81
	Dr. Michael Breu (Arctis): „Sicherheit und Web Service Architekturen“	Seite 95
	Christian Wied (beck et. al projects GmbH): „Konzeption einer serviceorientierten Veranstalter-Plattform für einen Reiseveranstalter“	Seite 107
Abschluss		
18:30 – 18:45	Diskussion	
Anschließend	Empfang	

2 Motivation und Einleitung

Service-orientierte Architekturen sind in aller Munde. Am deutlichsten wird dies, wenn man sich die Präsenz in den Medien kurz vor Augen führt. So stand in der Computerzeitung vom 11.7.2005 unter der Überschrift **Prozessorientierung geht vor Technik**: „...Damit bereiten sich die Firmen auf den Einzug serviceorientierter Architekturen (SOA) vor...“ Und so findet sich dieser Begriff quer verstreut über die gesamte Medienlandschaft wieder.

Doch was wollen Service-orientierte Architekturen eigentlich wirklich? Es ist mühsam hier eine vollständige Liste von Zielen aufzuzählen, da in verschiedenen Konstellationen Auftraggeber/Auftragnehmer, Organisation und Anbieter in der Regel immer andere Ziele verfolgt werden. Dies rührt sicherlich auch daher, dass die Ausgangssituationen verschieden sind. IT-Systeme sind heutzutage in der Regel *gewachsen*. Über einen langen Zeitraum wurden Investitionen in teure IT-Systeme getätigt, die zwar nicht mehr „up to date“ sind, jedoch immer noch die Aufgaben erfüllen, für die sie ursprünglich einmal beschafft wurden. Ist also eine Modernisierung das Ziel?

Für Entwickler und Dienstleister stellt sich die Situation anders dar: Technologien ändern sich; werden mächtiger und komplexer. Ist es also das Ziel, neue Technologien zu etablieren und sich in neu geschaffenen Märkten unter Zeit- und Kostendruck zu behaupten? Das sind Fragen, die gar nicht so neu sind. Auch die Zielstellungen sind nicht neu, haben durch die SOAs allerdings einen neuen Anstrich erhalten.

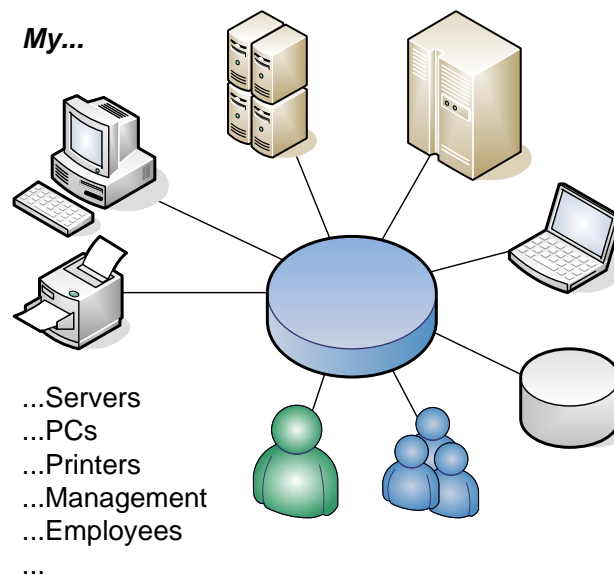


Abbildung 2.1: Vision: Einheitliche Sicht auf eine IT-Landschaft

Lediglich das Erreichen einer *Win/Win-Situation* scheint hier als Gemeinsamkeit zu existieren. Der IT-Dienstleister muss sich am Markt behaupten. Neue Technologien und dazu passende Expertisen sind hierzu unerlässlich. Das interessiert IT-Systembetreiber natürlich nicht in erster Instanz. Vielmehr stehen hier Erwartungen im Vordergrund, die hauptsächlich auf die Effizienz der IT-Landschaft abzielen und meistens die Komplexität beherrschbar machen sollen (vgl. Abbildung 2.1).

2.1 Komplexität von IT-Systemen und Landschaften

Was ist eigentlich Komplexität von IT-Systemen? Für die Komplexität kann man viele Maßstäbe ansetzen – ungünstigerweise sind diese auch noch miteinander *kombinierbar*. Einfache Beispiele für komplexe Strukturen in IT-Systemen sind:

- Technische Plattformen – Hardware: x86, PowerPC, Sparc, IA64 etc.
- Technische Plattformen – Software: Windows, Solaris, Linux etc.

Allein diese beiden Punkte sind schon fast frei kombinierbar, wenn man daran denkt, dass Windows beispielsweise sowohl für x86 als auch IA64 verfügbar ist und Linux sich auf so ziemlich jeder Plattform zuhause fühlt. Neben den eigentlichen Schlagworten kommen hier also auch *Varianten* hinzu, wobei man natürlich nicht von einer uneingeschränkten Kompatibilität ausgehen darf – ein 16Bit Windowsprogramm mag unter Windows XP noch funktionieren, unter Windows XP für 64Bit oder der Variante für IA64 kann das schon wieder ganz anders aussehen. Die Feststellung lautet also: Windows \neq Windows.

Dies mag noch nicht so beängstigend erscheinen, jedoch zeichnet die Realität ein noch viel radikaleres Bild: Typische Anwendungslandschaften großer Unternehmen sind sehr *heterogen* – **jede** verfügbare Plattform und **jedes** denkbare Betriebssystem sind hier irgendwo im Einsatz, wobei das *irgendwo* auch die Realität darstellt, da in vielen IT-Systemen of nicht bekannt ist, wie die aktuelle Konfiguration überhaupt aussieht. Damit müssen Unternehmen umgehen! Und dies setzt sich auch auf der Ebene der Anwendungssoftware fort. Ein neuer Begriff kommt hier ins Spiel: Die *Anwendungslandschaft*.

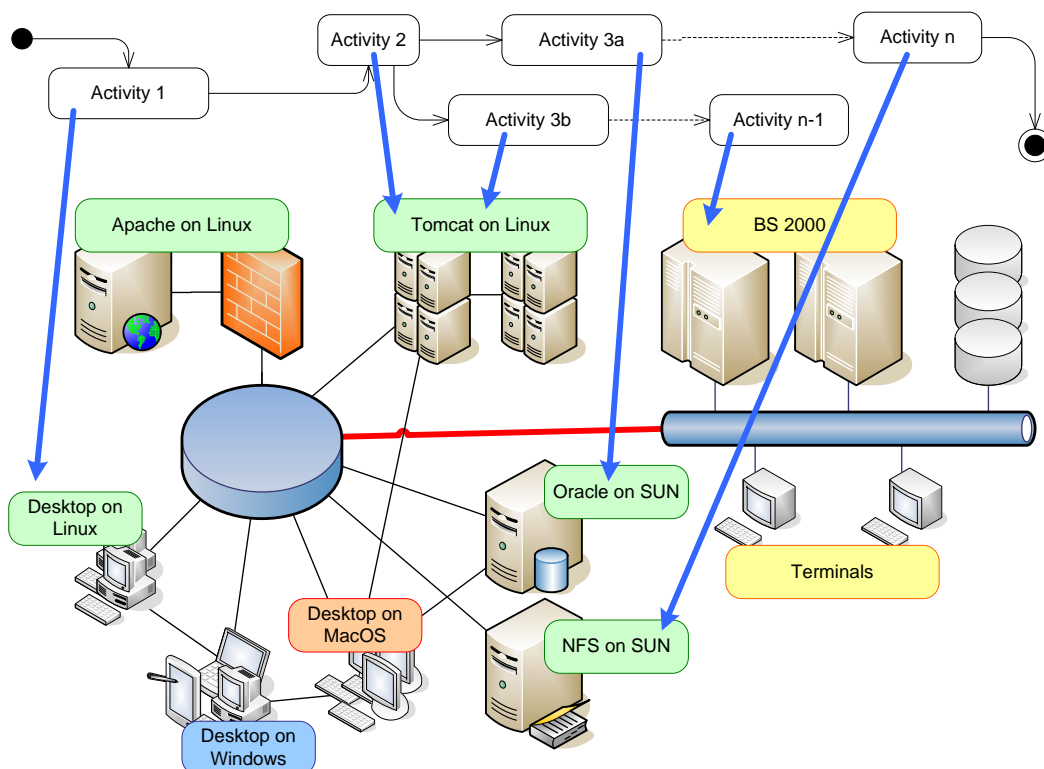


Abbildung 2.2: Vernetzte heterogene IT-Landschaft

Es ergeben sich Systeminfrastrukturen, die am einfachsten noch als kombinatorisches Problem über Mengen von Hardwarekomponenten, Softwarebasiskomponenten und Anwendungssoftwarekomponenten zu bezeichnen sind. Alle Kom-

ponenten sind in der Regel bereits in irgendeiner Form vorhanden (vgl. Abbildung 2.2) und funktionieren mehr oder weniger gut zusammen. Einheitliche Konzepte zum Betrieb, zur Wartung und zur Weiterentwicklung sind Mangelware.

2.2 Vernetzung als Komplexitätstreiber

Verschärft wird die ohnehin schwierige Ausgangssituation noch durch die zunehmende Vernetzung (siehe Abbildung 2.2). IT-Systeme haben in der Regel sehr viele fachliche und technische Schnittstellen. Zusammen wirkend werden Geschäftsprozesse abgebildet. Diese und fachliche Capabilities sind dabei über die gesamte Anwendungslandschaft verstreut.

Eine einheitliche Sicht lässt sich über einer derart komplexen Landschaft mit einigem Aufwand sicher noch definieren. Doch wenn es dann um die Integration zusätzlicher Systeme geht wird dies zu einem sehr aufwändigen Unterfangen. Probleme bei Wartung und Betrieb eines solchen Systems müssen an dieser Stelle nicht noch einmal explizit hervorgehoben werden.

2.3 Unternehmensdynamik als Komplexitätstreiber

Doch es gibt auch eine neue Problemklasse, die hier mit in Betracht gezogen werden muss – Unternehmensfusionen. Bei einer Fusion von zwei Unternehmen treffen mindestens zwei, unter Umständen völlig inkompatible IT-Landschaften aufeinander, die integriert werden müssen (vgl. Abbildung 2.3).

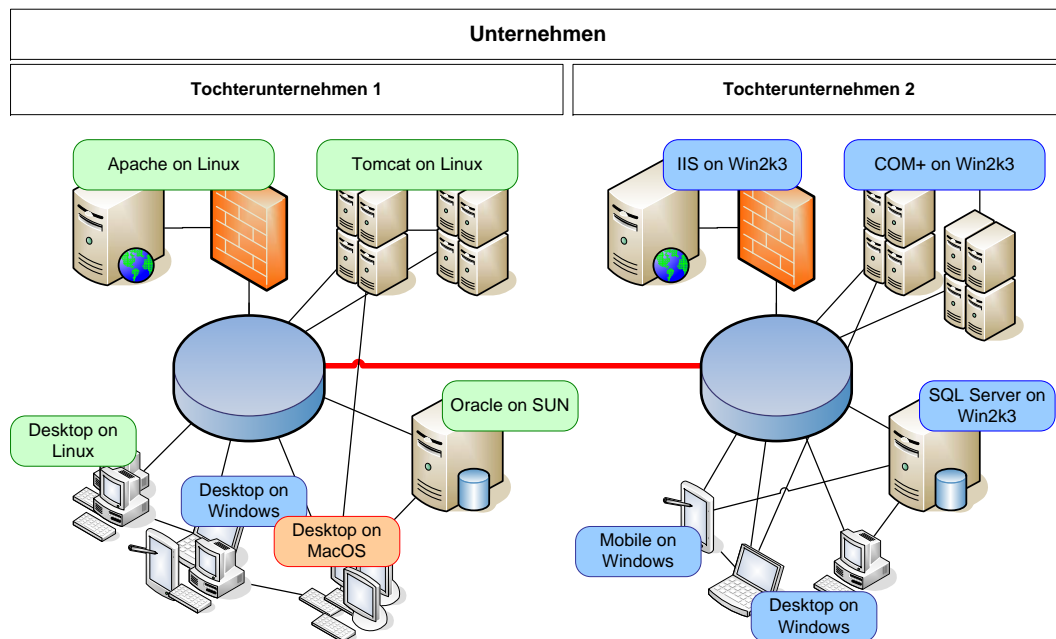


Abbildung 2.3: Integrationsproblem von IT-Landschaften bei Unternehmensfusionen

Neben Struktur- und Kommunikationsaspekten müssen hier noch andere Punkte beachtet werden. Unter der Annahme, dass beide Unternehmen etablierte Rechnerunterstützte Prozesse für das Bestell- oder Rechnungswesen haben: Welches davon wird verwendet? Nur eines? Wenn ja, welches? Beide? Wie sind die dann zu integrieren, da es sich ja nun um einen ganzheitlichen Prozess handeln muss? Keines? Wird also ein komplett neues System aufgesetzt? Wie sind dann die grundlegenden Prozesse und organisatorischen Vorgaben derart umzusetzen, damit keiner der

beiden Unternehmensteile unberücksichtigt bleibt? Und so weiter. Interessant an dieser Fragestellung ist natürlich, wenn man sie weiterdenkt und sich dabei große Institutionen (egal ob im industriellen oder öffentlichen Umfeld) ansieht, dass diese Problematik oft schon inhärent in einer einzigen Organisation existiert.

2.4 Was wollen SOA's?

IT-Systeme sind komplex. Die Komplexität erstreckt sich über mehrere Ebenen hinweg beginnend bei kleinsten technischen Fragestellungen bis hin zur Managementebene. SOAs haben den Anspruch, einen Weg aufzuzeigen, der die Komplexität von IT-Systemen beherrschbar macht. Die Erwartungen und Wünsche an SOAs sind hoch:

- Reduktion der technischen Heterogenität durch einheitliche Protokolle
- Reduktion der Wartungskosten durch Unabhängige Dienste
- Verbesserung der Wiederverwendung fachlicher Dienste
- ...

Viele Software- und Beratungshäuser stehen momentan mit Angeboten am Markt. So existieren (technische) Lösungen für Anwendungsintegration, Prozessmodellierung und -orchestrierung. Weiterentwicklungen von Laufzeitsystemen versprechen höhere Abstraktion und Integration via Nachrichtenaustausch. Beratungshäuser bieten Leistungen im Bereich Beratung und Coaching rund um die Einführung einer SOA an. Die eigentlichen Probleme werden jedoch nur vereinzelt adressiert (z. B. die Suche nach zeitlich stabilen Unternehmenseigenschaften, klare Ziel- und Kostendefinition oder die Ausarbeitung einer IT-Strategie). Kurzfristige Lösungen führen dabei nicht im gewünschten Maße zum Ziel („quick wins“ sind selten), sodass oftmals auf Neuentwicklungen oder Erweiterungsentwicklungen ausgewichen wird. Die Neuentwicklung von IT-Systemen kann aber zunächst unberücksichtigt bleiben, da die existierende Komplexität schon alle kritischen Eigenschaften besitzt:

- Komplexität ist aufwändig, zum Beispiel im Betrieb, bei der Wartung etc.
- Komplexität erhöht die Wahrscheinlichkeit von Fehlern. In einem global agierenden Unternehmen muss ein Fehler, der beispielsweise in Hamburg auftritt, nicht auch dort produziert worden sein, sondern kann sich beispielsweise von San Francisco über London fortgepflanzt haben.
- Komplexität ist **teuer**.

Im Wesentlichen erwartet man sich also eine Verbesserung des Ist-Zustands hinsichtlich der Beherrschbarkeit der etablierten IT-Landschaft, ohne die eine Erweiterung, beziehungsweise eine Restrukturierung wenig sinnvoll ist. Man erwartet sich Möglichkeiten zur Systemintegration in Geschäftsfeldern, in denen Unternehmen zu koppeln sind (B2B), beziehungsweise der Kunde näher an ein Unternehmen herangeholt werden soll (B2C). Dies alles zieht natürlich in gewissem Maße eine Neuausrichtung hinsichtlich Modernisierung und Interoperabilität nach sich.

Doch wie so oft löst SOA nicht nur Probleme, sondern wirft auch neue auf, die im Folgenden anhand einfacher kurzer Fragestellungen aufgezeigt werden sollen:

- Was ist eigentlich ein Service? Eine Funktion oder eine Anwendung oder ein System?

- Was ist dann eine Service-orientierte Architektur? Sind Service-orientierte Architekturen überhaupt Architekturen oder beschreiben die geforderten Eigenschaften lediglich Konzepte?
- Sind Service-orientierte Architekturen dann ein Prozessthema? Wodurch grenzen sie sich dann von den EAI-Bestrebungen ab? Was ist dann ein Service im Sinne eines Prozesses? Wie stabil sind Prozesse? Sind sie eine Grundlage für eine tragfähige Architektur?
- Welche Rollen spielen Web Services? Müssen Web Services verwendet werden, um Service-orientiert zu sein? Ist alles andere hinfällig? Gilt mit Web Services: „one size fits all“?
- Was ist wirklich neu? Unterscheidet sich die Komponentenkopplung wirklich so sehr von der Anwendungskopplung?

2.5 Der Workshop

Wie im letzten Abschnitt zu sehen war, existieren viele offene Fragen, die zu beantworten sind. Im Rahmen des Workshops Service-orientierte Architekturen wurden diese Fragen gestellt und zum Teil kontrovers diskutiert.

Eröffnung

Herr Prof. Broy hob hierzu die Bedeutung des Dienstbegriffs hervor. Über den Hype hinausgehend ist er auch für die Forschung von großem Interesse. Des Weiteren betonte er gerade für diesen Bereich die Wichtigkeit der Kommunikation und des Gedankenaustauschs zwischen Forschung und Industrie.

Service-orientierte Architekturen – Strategien und Visionen

Herr Rohe von der Microsoft Deutschland GmbH eröffnete die Vortragsserie, indem er sich zu Visionen und Strategien im Umfeld Service-orientierter Architekturen äußerte. Er stellte Kernprobleme wie Kopplung von Komponenten und den zweifelhaften Nutzen der Prozessorientierung bei der Ableitung von Services zur Diskussion. Weiterhin schlug eine Methode vor, die in diesem Kontext ein viel versprechender Ansatz ist und schloss mit einer Variante einer möglichen Werkzeugunterstützung.

Web Services – Ideale Plattform oder Mittel zum Zweck?

Herr Dr. Haller von der sd&m AG referierte über die Fragestellung, ob denn Web Services nun eigentlich eine ideale Plattform für Service-orientierte Architekturen seien oder einfach nur ein Mittel zum Zweck. Er stellte Thesen zu Web Services auf und benannte Stärken, Schwächen, Chancen und Risiken.

SOA – Von der Vision zur Realität

Herr Popescu von HVB Systems GmbH berichtete über erste Erfahrungen bei Versuchen einer Umsetzung einer Service-orientierten Architektur. Er stellte dar, dass es wohl extreme Ansätze gibt, die immer irgendwem nützen, von denen aber keiner wirklich praktikabel sei. Er hob außerdem noch einmal die unbedingt notwendige Unterstützung des Managements hervor, da SOA ein ganzheitlicher Ansatz für mindestens ein Unternehmen ist.

SOA – Zwischen Anspruch und Wirklichkeit

Herr Josuttis (IT Communication) berichtete ernüchtert von seinen Erfahrungen und führte Fragen auf, die die Anwendbarkeit der SOA hinterfragen. So zum Beispiel stellte er die Fragen zur Diskussion wie denn die Granularität von Services zu bestimmen sei oder wie man Stabilität gegenüber Flexibilität abwägen soll. Er schloss mit der Feststellung, es scheine so, als würden sich dieselben Fehler wie bei der Einführung der Objektorientierung wiederholen.

Wann liefert eine Serviceorientierte Architektur echten Nutzen?

Herr Dr. Richter (sd&m AG) referierte über die Frage, wann denn SOA überhaupt einen Nutzen bringt. Er stellte fest, dass SOAs keine Technikprojekte seien sondern im weitesten Sinne Strategien, welche die gesamte IT-Landschaft betreffen und dass auch die Unterstützung des Managements erforderlich sei.

Wozu SOA?

Herr Dr. Pizka (TUM) stellte in seinem Vortrag kritische Fragen, unter anderem zum wirklichen Fortschritt, der mit Service-orientierten Architekturen erreicht wird. Er stellte fest, dass die Ausrichtung an Geschäftsprozessen zur Identifikation von Services eine kritische Praxis ist, die zumeist potenziell instabile Systeme erzeugt.

Modellbasierte dienstorientierte Software-Entwicklung

Herr Grünbauer (TUM) gab einen Einblick in die Arbeiten zur modellbasierten Dienstentwicklung im Automotivbereich und verwies auf die hohe Komplexität derartiger Systeme und dass eine korrekte und fundierte Modellierung zwingend seien.

SOA – aber wie?

Frau Hanschke (iteratec GmbH) präsentierte die Ansichten und das Verständnis von iteratec zum Thema SOA. Sie präsentierte ein Konzept, das Prinzipien und Leitfäden umfasst und stellte heraus, dass eine SOA erfolgreich sein kann, wenn bereits im Vorfeld Grundlagen dafür geschaffen werden.

Sicherheit und Web Servicearchitekturen

Herr Dr. Breu (Arctis) befasste sich mit der Sicherheit in Web Service Architekturen und identifizierte dieses Thema als Querschnittsthema, das auch im Kontext Service-orientierter Architekturen große Relevanz besitzt.

Konzeption einer SOA Veranstalter-Plattform

Herr Wied (beck et. al project GmbH) präsentierte abschließend ein Konzept einer Service-orientierten Plattform, stellte Motivation und eine Referenzarchitektur hierfür vor.

2.6 Die Essenz

Der Workshop bot ein Programm großer Breite. Es bestand Diskussionsbedarf; der Leser möge sich hierzu seine Meinung bilden. Insgesamt ist das Fazit, dass aus diesem Workshop gezogen werden kann, jedoch ernüchternd.

Es konnte festgestellt werden, dass die Marketingmaschinerie in den vergangenen Jahren sehr aktiv war und eine beachtliche Anzahl einschlägiger, verkaufsträchtiger Schlagwörter generiert hat. Wie so oft muss man in Praxis jedoch feststellen, dass es meistens ganz anders läuft, wie z. B. Herr Jossutis festhielt. Auch die medienpräsenten Big Player wie Microsoft erkennen, dass andere Wege beschritten werden müssen. Wie Herr Rohe, unterstützt von Herrn Pizka, feststellte, sind Geschäftsprozesse nicht immer die beste Ausgangsbasis für die Ableitung fachlicher Services. Die skizzierte Methode MOTION verspricht einiges an Potenzial.

Herr Haller und Herr Richter von sd&m stellten noch einmal fest, dass SOAs prinzipiell keine Technikprojekte sind, sondern strategische Vorhaben, die nicht nur auf die IT, sondern auf eine ganze Organisation wirken. Wenn eine SOA dann doch an einer Technik festgemacht wird, so stellte Herr Haller fest, müssen es nicht unbedingt Web Services sein. Dies begünstigt lediglich das sich schon seit längerem haltende Gerücht, das Web Service Architekturen und SOAs dasselbe seien. Auch hier sind wieder Chancen mit Risiken abzuwägen. Unterm Strich schien es auch so, dass die Einführung einer SOA nicht durch bloßes Durchführen eines Pilotprojekts bewältigt werden kann, sondern eher mit der Einführung eines Vorgehensmodells vergleichbar ist. Die Argumente sind ähnlich:

- „Quick wins“ sind selten.
- Es dauert eine gewisse Zeit, sodass Unternehmen, die sich zu diesem Schritt entschließen, einen entsprechend langen Atem benötigen.
- SOAs betreffen nicht nur die IT, sondern das gesamte Unternehmen, also auch die Mitarbeiter.

Sind SOAs nun also ebenso nur ein Hype wie die EAI-Bestrebungen? Das ist zu verneinen, da man sich den sich stetig ändernden Märkten anpassen muss. Diese Anpassungen finden heute zu großen Teilen in Form von Unternehmensfusionen auf der einen und Outsourcing auf der anderen Seite statt. Fakt scheint jedoch, dass eine Phase der größeren Ernüchterung kommen wird, wenn gerade die großen Anbieter Farbe bekennen und darstellen müssen, was denn nun wirklich neu ist und wo der Mehrwert einer SOA ist.

Letztendlich wird sich die SOA zumindest auf der kommunikativen Ebene zwischen Anbietern und Entscheidern durchsetzen. Ob sie den Erwartungen gerecht wird bleibt weiterhin abzuwarten.

Service-orientierte Architekturen – Strategien und Visionen

Klaus Rohe (klrohe@microsoft.com)
Developer Platform & Strategy Group
Microsoft Deutschland GmbH

1

Microsoft

Agenda

- ❖ Motivation für SOA
- ❖ SOA Grundprinzipien
- ❖ Mehrwert einer SOA aus Sicht der
Geschäftsanforderungen
 - ◆ Anwendungsentwicklung und Unternehmensarchitektur
 - ◆ Business Capability Modell
- ❖ SOA auf der Microsoft Windows Plattform
 - ◆ Servertechnologie
 - ◆ Clienttechnologie
 - ◆ Tools zur Modellierung und Implementierung einer
SOA
 - ◆ Ausblick
- ❖ Zusammenfassung

2

Microsoft

Agenda

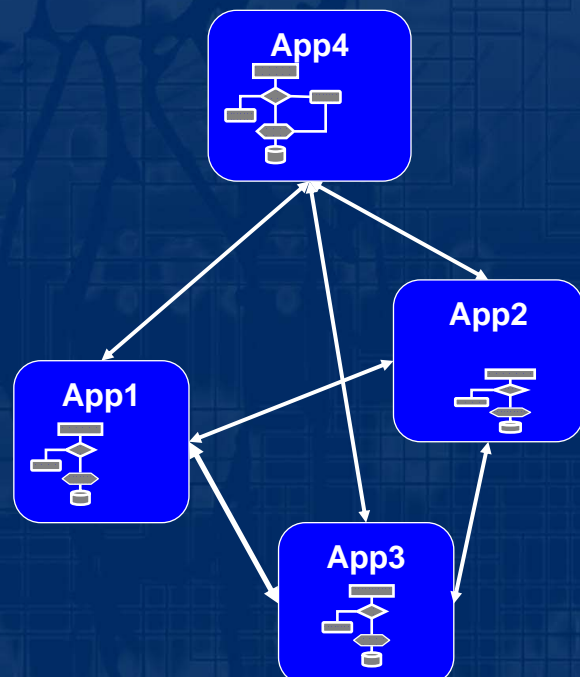
- ❖ Motivation für SOA
- ❖ SOA Grundprinzipien
- ❖ Mehrwert einer SOA aus Sicht der Geschäftsanforderungen
 - ♦ Anwendungsentwicklung und Unternehmensarchitektur
 - ♦ Business Capability Modell
- ❖ SOA auf der Microsoft Windows Plattform
 - ♦ Servertechnologie
 - ♦ Clienttechnologie
 - ♦ Tools zur Modellierung und Implementierung einer SOA
 - ♦ Ausblick
- ❖ Zusammenfassung

3

Microsoft

Die aktuelle Situation

- ❖ Geschäftsprozesse sind in Applikationen fest codiert:
 - ♦ In monolithischen Applikationen (ERP, CRM, ...)
 - ♦ Verteilt über mehrere Applikationen (EAI)
- ❖ Resultat ist eine hohe Komplexität der Applikationsarchitektur.
 - ♦ $n * (n - 1) / 2$ Schnittstellen im schlimmsten Fall

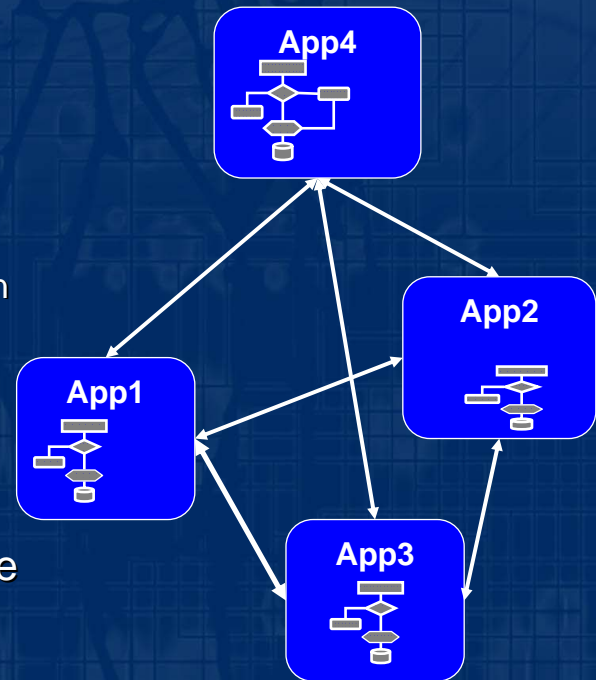


4

Microsoft

Das resultierende Problem: keine Flexibilität

- ❖ Fakt: Geschäftsprozesse ändern sich häufig
- ❖ Anpassungen der Software ist sehr schwierig:
 - ◆ Schnittstellen sind über viele Applikationen verteilt.
 - ◆ Abhängigkeiten der Applikationen untereinander bewirken hohe Fehlerrate bei Änderungen und Anpassungen.
- ❖ Anpassung von Software an geänderte Geschäftsprozesse ist sehr zeitintensiv und teuer.
- ❖ Wie können Geschäftsprozesse flexibler durch Software abgebildet werden?

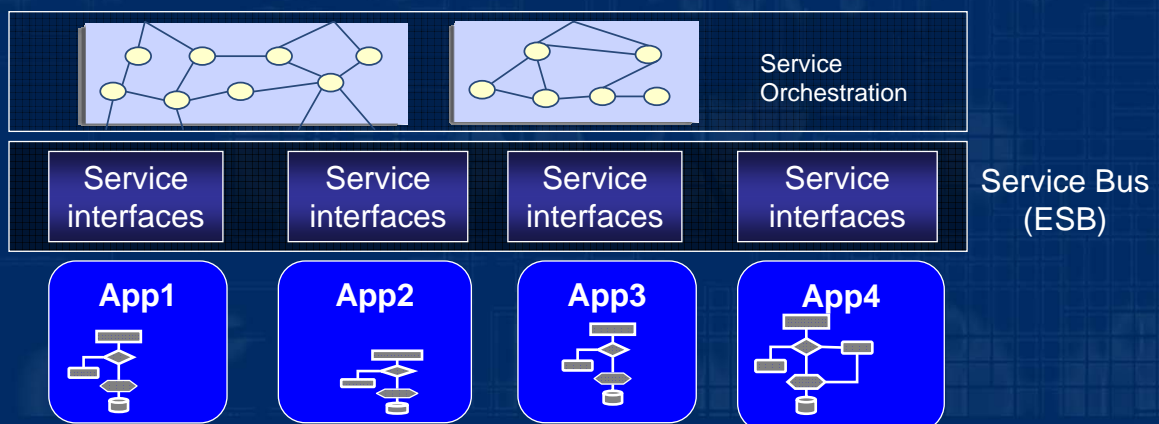


5

Microsoft

Lösung: Service-orientierte Architektur (SOA)

- ❖ Geschäftsprozesse können als Orchestrierung von Services abgebildet werden.
- ❖ Orchestrierung wird durch eine „Sprache“ mit adäquaten Abstraktionsniveau beschrieben.
- ❖ Schnittstelle der Services muss eine hohe zeitliche Stabilität haben.
 - ◆ *Wie findet man solche Services?*



6

Microsoft

Agenda

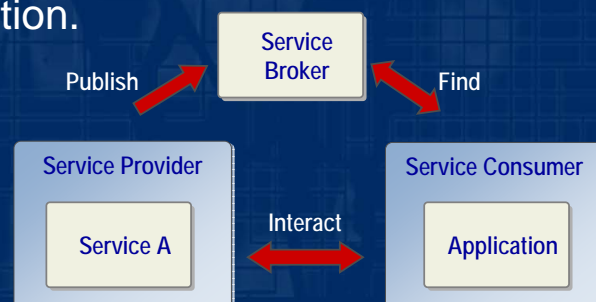
- ❖ Motivation für SOA
- ❖ SOA Grundprinzipien
- ❖ Mehrwert einer SOA aus Sicht der Geschäftsanforderungen
 - ◆ Anwendungsentwicklung und Unternehmensarchitektur
 - ◆ Business Capability Modell
- ❖ SOA auf der Microsoft Windows Plattform
 - ◆ Servertechnologie
 - ◆ Clienttechnologie
 - ◆ Tools zur Modellierung und Implementierung einer SOA
 - ◆ Ausblick
- ❖ Zusammenfassung

7

Microsoft

Service Oriented Architecture

- ❖ “A Service-Oriented Architecture (SOA) breaks applications into large granularity service components, whose interactions may be bi-directional and asynchronous and are governed by properly specified message-based contracts ...” *Jack Greenfield, Keith Short **Software Factories***
- ◆ Loosely coupled distributed components and applications.
- ◆ Standardized, message based communication protocol
- ◆ SOA anticipates integration.

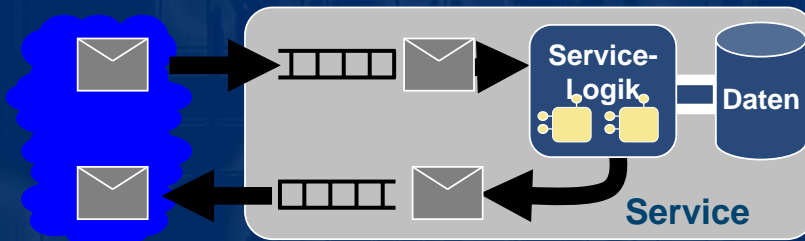


8

Microsoft

SOA im Kontext von Softwaretechnologie

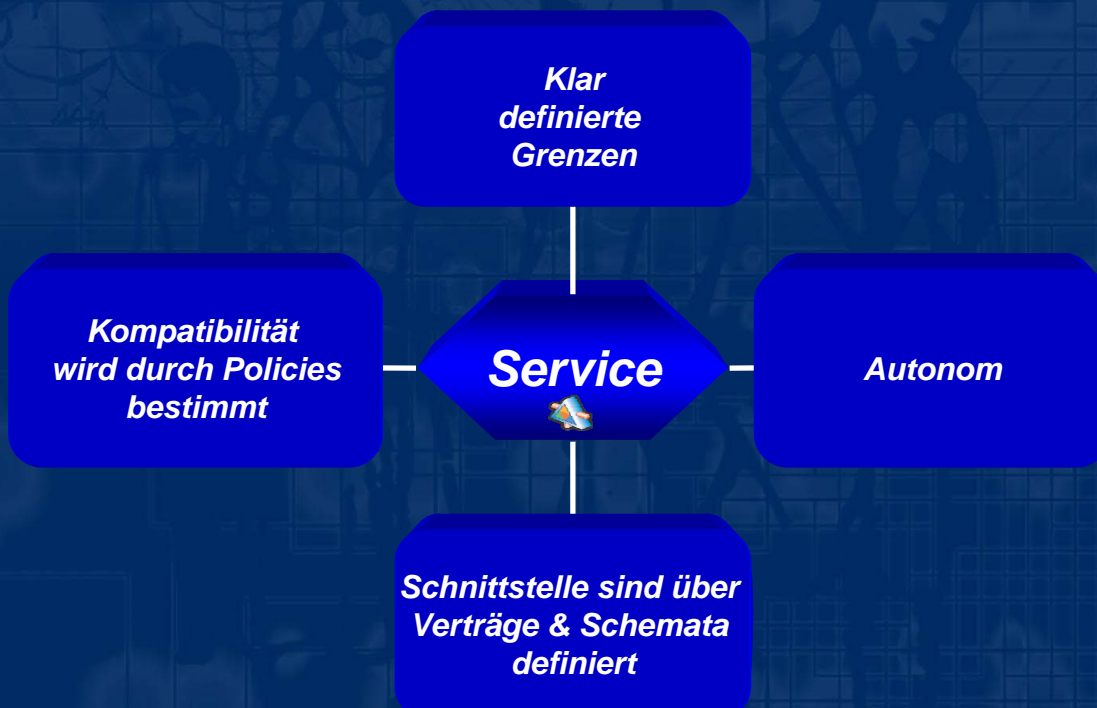
- ❖ SOA ist kein Ersatz für die Objekttechnologie
- ❖ Nutzt bewährte Erkenntnisse und Methoden aus den Bereichen
 - ◆ TP-Monitore, OTMs
 - ◆ Message-Oriented-Middleware (MOM)
 - ◆ Komponententechnologie
 - ◆ verteilte Objekttechnologie
- ❖ Neu:
 - ◆ Nutzung von plattformunabhängigen und internet-fähigen Kommunikations- und Servicemodellen



9

Microsoft

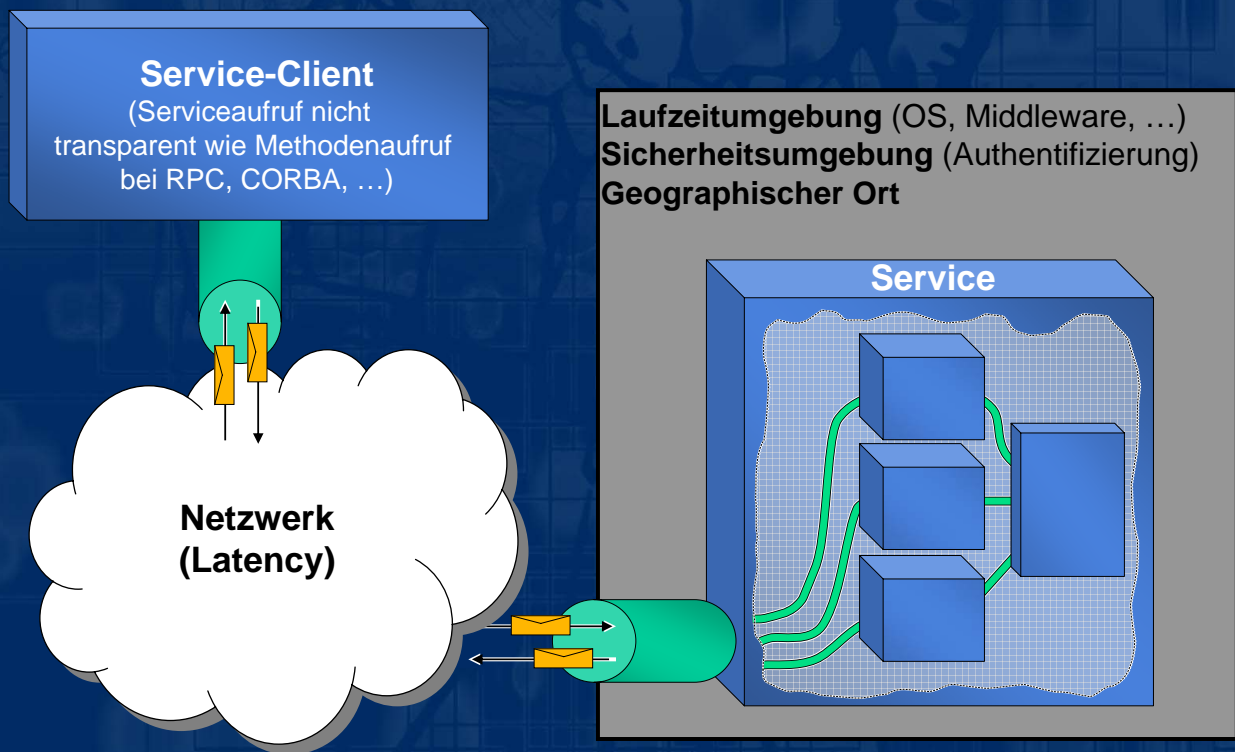
Die vier Grundprinzipien für Services



10

Microsoft

Ein Service hat klar definierte Grenzen



11

Microsoft

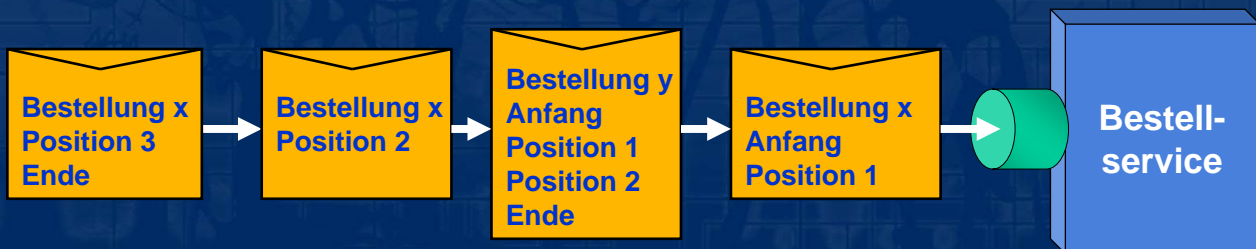
Services sind autonom

- ❖ Kein übergeordneter Mechanismus, der die Services kontrolliert.
- ❖ Ein Service bietet eine in sich geschlossene Dienstleistung an.
- ❖ Die Umgebung in welcher ein Service existiert, ist nicht statisch.
 - Services in der Umgebung, die genutzt werden, können ihre Implementierung ändern.
 - Services die genutzt werden können geografisch verlagert werden oder können verschwinden.
- ♦ Dies darf keine Auswirkungen auf den Nutzer haben

12

Microsoft

Serviceschnittstellen sind über Verträge und Schemata definiert



- ❖ Der Vertrag beschreibt die zulässige Reihenfolge der Messages („Applikationsprotokoll“).
- ❖ Die Struktur der vom Service akzeptierten Messages wird über ein Schema beschrieben.
- ❖ Vertrag und Schema bleiben zeitlich konstant!

13

Microsoft

Kompatibilität von Services wird über Policies definiert

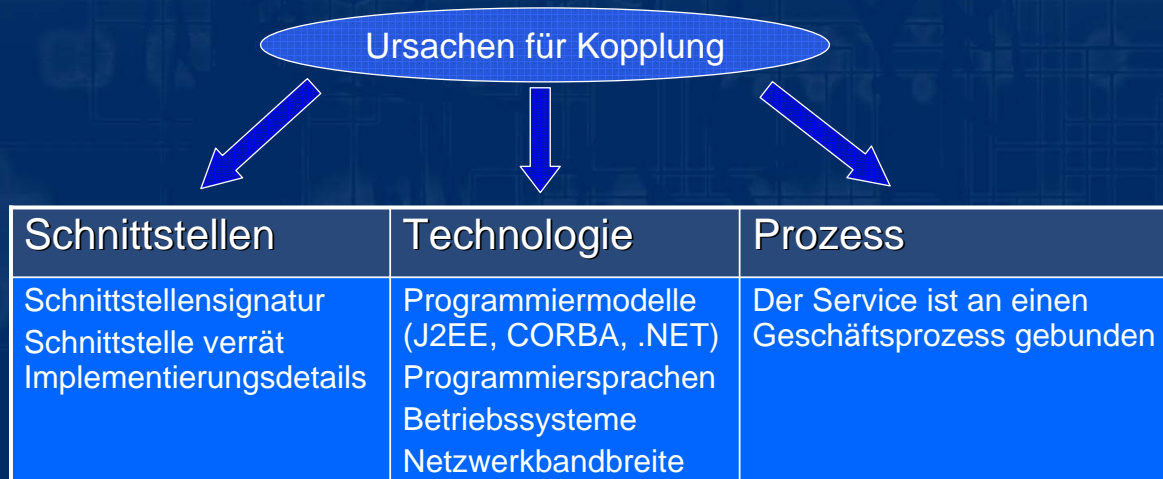
- ❖ Ein Service veröffentlicht die Richtlinien und Verfahrensweisen, Policies genannt, wie er zu nutzen ist.
 - ◆ Richtlinien, die „maschinenlesbar“ sind.
 - ◆ Beschreiben Anforderungen und Leistungen des Services
- ❖ Policies definieren, welche Bedingungen erfüllt sein müssen, damit eine normale Operation des Services gewährleistet ist.
- ❖ Hinweis: WS-Policy

14

Microsoft

Serviceorientierung und Kopplung

- ❖ Kopplung zwischen Client und Service:
 - ♦ Anzahl der Voraussetzungen, die eine Client erfüllen muss, um einen Service zu nutzen.
- ❖ Je kleiner die Anzahl der geforderten Voraussetzungen, desto loser die Kopplung.



15

Microsoft

Web-Services unterstützen lose Kopplung auf der Ebene der Technologie und Schnittstellen

- ❖ Unabhängig von:
 - ♦ Hardware
 - ♦ Betriebssystem
 - ♦ Middleware
 - ♦ Programmiersprachen und Programmiermodellen.
- ❖ Geeignet für das Internet.
 - ♦ Service kann an jedem Ort genutzt werden, an dem Internetzugang vorhanden ist.
 - ♦ Keine besonderen Anforderungen an Netzwerkbandbreite.
- ❖ Nachrichten basierte, zuverlässige Kommunikation möglich.
 - ♦ Keine Abhängigkeit von Methodensignaturen.
- ❖ **Web-Services und die zugehörigen Standards / Spezifikationen sind die technologische Basis auf der man heute eine SOA realisieren sollte.**

16

Microsoft

Agenda

- ❖ Motivation für SOA
- ❖ SOA Grundprinzipien
- ❖ Mehrwert einer SOA aus Sicht der Geschäftsanforderungen
 - ♦ Anwendungsentwicklung und Unternehmensarchitektur
 - ♦ Business Capability Modell
- ❖ SOA auf der Microsoft Windows Plattform
 - ♦ Servertechnologie
 - ♦ Clienttechnologie
 - ♦ Tools zur Modellierung und Implementierung einer SOA
 - ♦ Ausblick
- ❖ Zusammenfassung

17

Microsoft

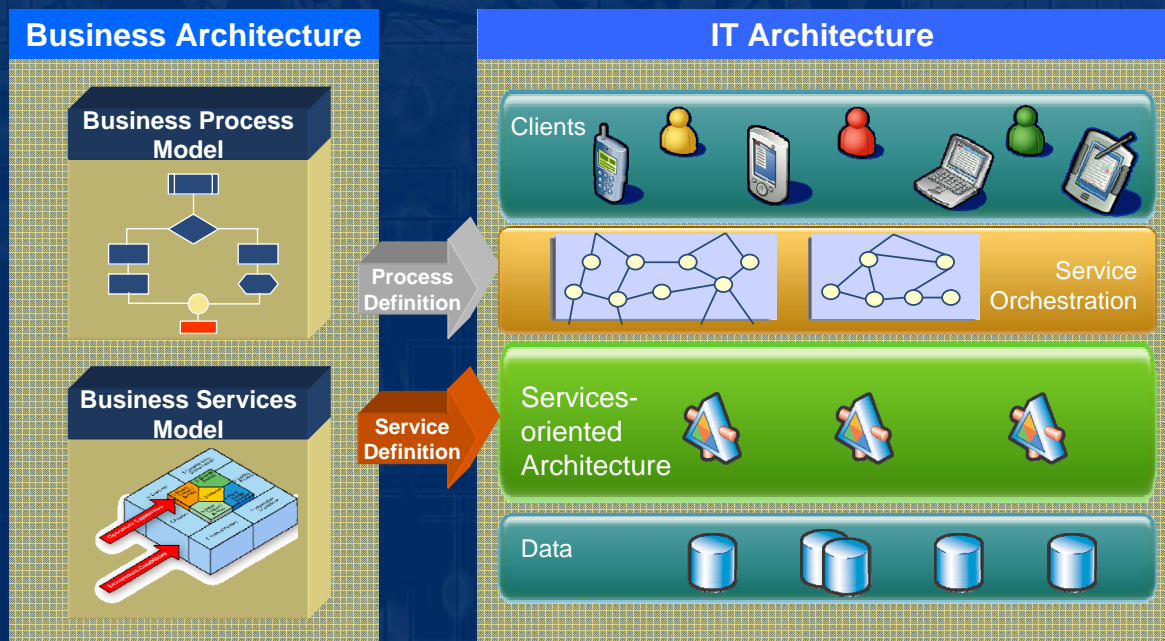
Mehrwert einer SOA

- ❖ Softwareanwendungen werden als Komposition (Orchestrierung) von **Services implementiert, die eine stabile (zeitlich betrachtet) Schnittstelle anbieten.**
- ❖ Komposition (Orchestrierung) der **Services** kann einfach und schnell geändert und angepasst werden. Dadurch zeitnahe Reaktion auf Änderungen in den Geschäftsprozessen.
- ❖ „Services als Applikationsbausteine.“
- ❖ „Bessere Entkopplung der Geschäftsprozesse von ihrer softwaretechnischen Implementierung.“
- ❖ Entscheidende Frage:
 - ♦ Wie findet man Services deren **Schnittstelle zeitlich stabil ist?**

18

Microsoft

SOA in einer Unternehmensarchitektur

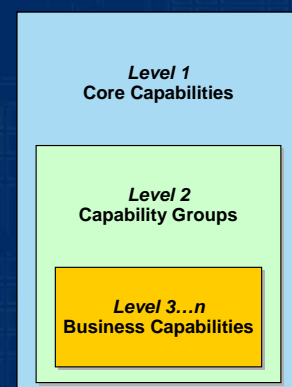
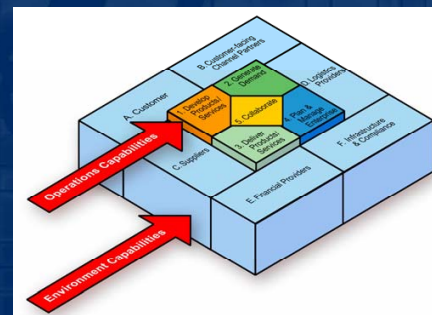


19

Microsoft

Prozess- und Organisationssichten sind instabil „Capabilities“ (Fähigkeiten) sind sehr stabil

- ❖ Ein Geschäftsprozess beschreibt detailliert wie etwas ausgeführt wird einschließlich der Ressourcen.
- ❖ Es ist allgemein üblich, dass Geschäftsprozesse häufig geändert bzw. angepasst werden.
- ❖ Wir glauben das eine Sicht unter dem Aspekt der Unternehmens-„Capabilities“ (Fähigkeiten) eine stabilere Basis liefert als die Prozesssicht.
- ❖ Definition „Business Capability“: eine spezifische Fähigkeit eines Unternehmens ein bestimmtes Ergebnis zu erzielen.
 - ♦ Beispiel: Auftrag erfassen
- ❖ Ein „Business Capability Model (BCM)“ ist eine hierarchische Anordnung von „Business Capabilities“.

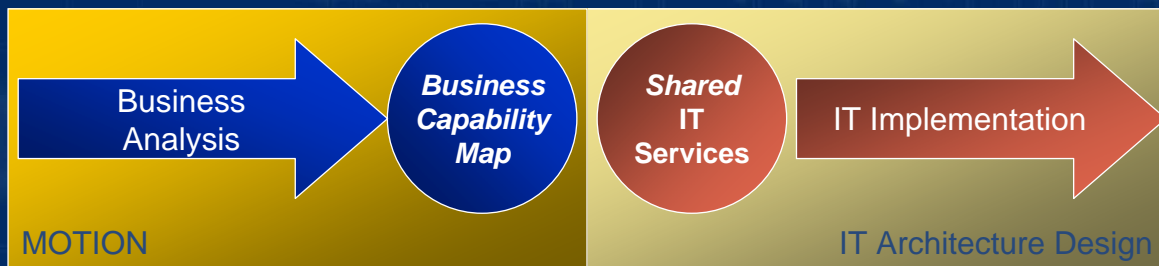


20

Microsoft

MOTION eine Methode zur Entwicklung einer „Business Capability Map“

- MOTION ist eine strukturierte Methode, die von Schlüsselmetriken ausgeht und eine „Business Capability Map“ des aktuellen Unternehmenszustands entwickelt.
- MOTION leitet aus einem Review der „Business Capability Map“ Aktionen und Empfehlungen für die Gestaltung von Geschäftsprozessen und IT ab.
- Die Ergebnisse von MOTION können als Basis für die Kommunikation von Geschäftsanforderungen an die IT genutzt werden.



21

Microsoft

Agenda

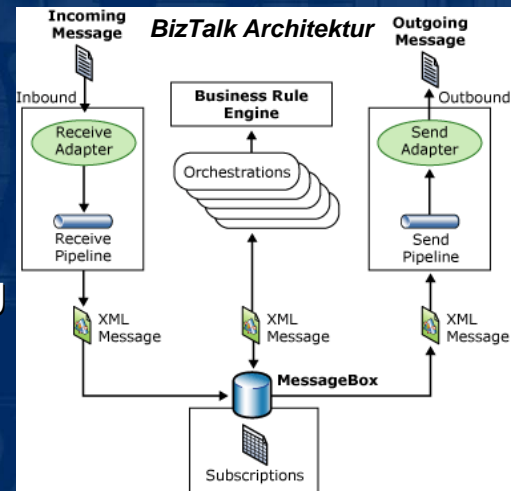
- ❖ Motivation für SOA
- ❖ SOA Grundprinzipien
- ❖ Mehrwert einer SOA aus Sicht der Geschäftsanforderungen
 - ◆ Anwendungsentwicklung und Unternehmensarchitektur
 - ◆ Business Capability Modell
- ❖ SOA auf der Microsoft Windows Plattform
 - ◆ Servertechnologie
 - ◆ Clienttechnologie
 - ◆ Tools zur Modellierung und Implementierung einer SOA
 - ◆ Ausblick
- ❖ Zusammenfassung

22

Microsoft

BizTalk Server 2004

- ❖ BizTalk Server 2004
 - ♦ Für komplexe Messaging-Integrations- und Transformationsaufgaben (EAI, B2B)
 - ♦ Content Based Routing (CBR)
 - ♦ Komposition und Orchestrierung von Web-Services, unterstützt BPEL
 - ♦ Metadaten getrieben
- ❖ BizTalk Server 2004 ist Microsofts aktuelles Werkzeug zur Implementierung eines Enterprise Service Bus (ESB).



23

Microsoft

Microsoft Office Information Bridge Framework (IBF)

- ❖ Bringt die Vorteile der SOA zum Endnutzer.
- ❖ IBF gibt Softwareentwicklern ein einfaches Modell an die Hand, mit dem sie Clientapplikationen erstellen können, die dem Benutzer eine einheitliche Sicht auf geschäftskritische Applikationen, wie z.B. ERP- und CRM-Systeme bieten.
- ❖ Metadaten getriebenes Framework, zum einfachen Zugriff auf Web-Services.
- ❖ IBF bietet einen deklarativen Ansatz zum Konstruieren von Applikationen, die auf den folgenden drei Komponenten basieren:

Web
Services

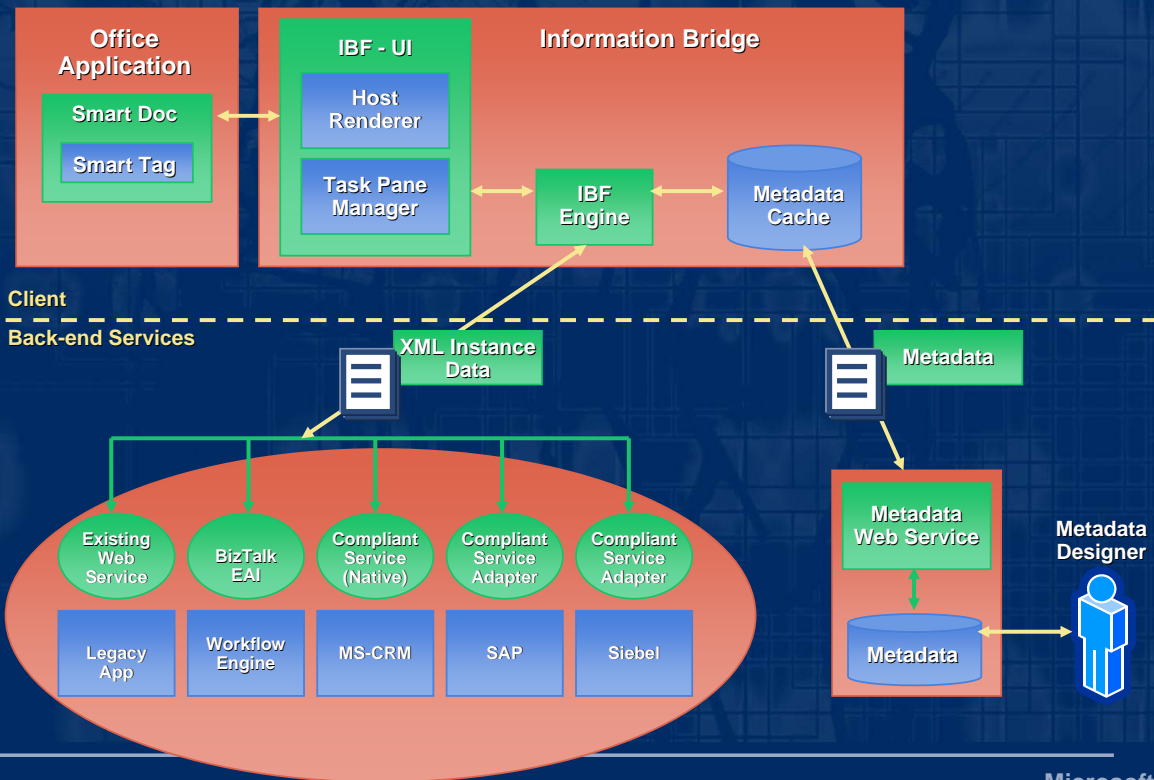
Metadata

Office Add-in
[Supports Excel,
Outlook, Word]

24

Microsoft

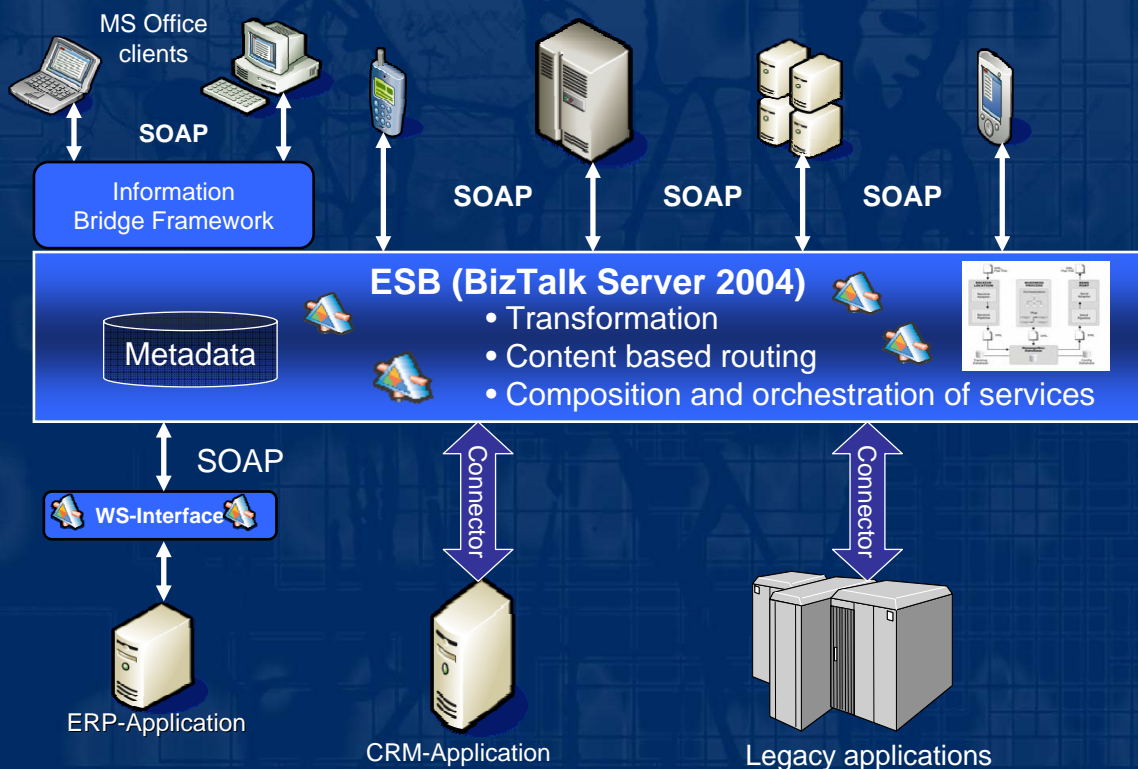
IBF Architekturüberblick



25

Microsoft

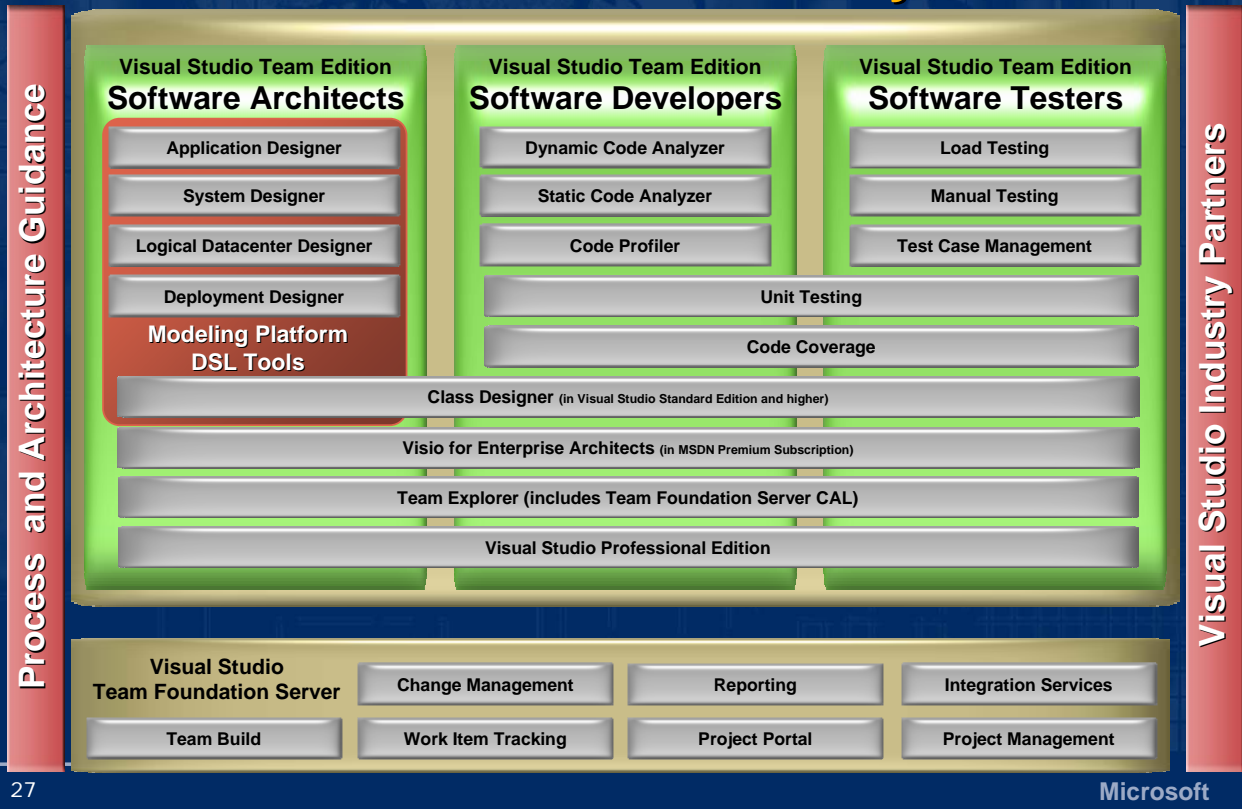
SOA auf der Microsoft Plattform



26

Microsoft

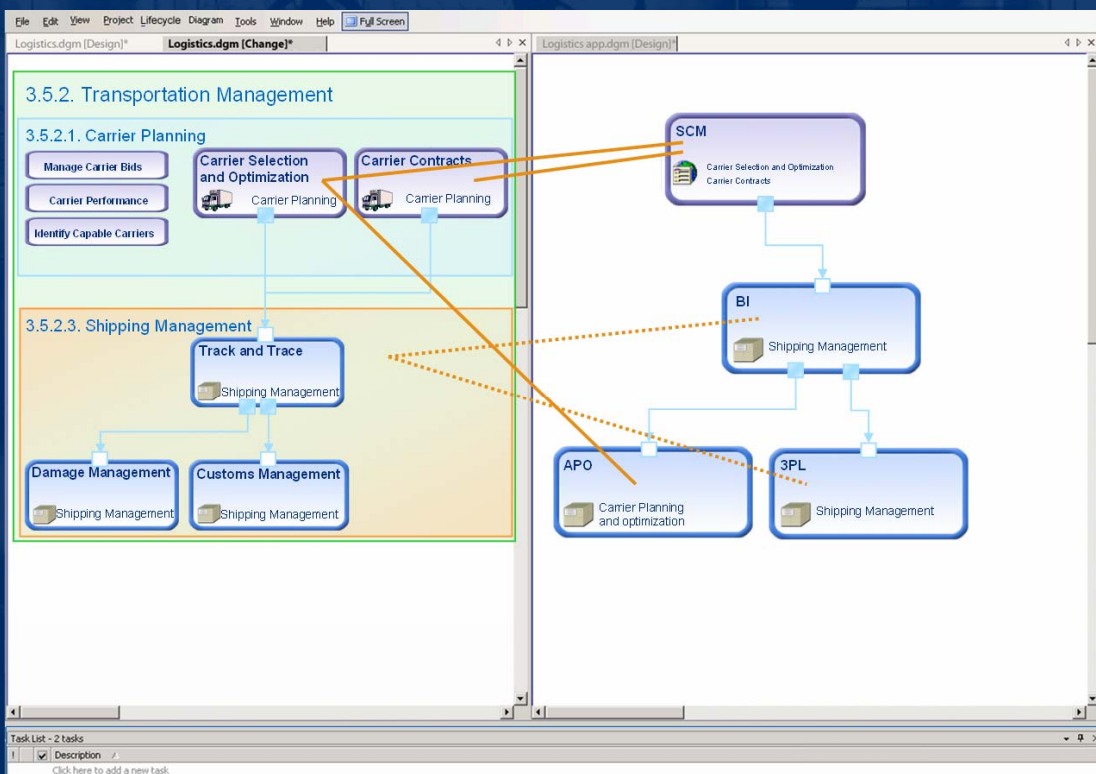
Tools zur Modellierung und Implementierung einer SOA: Visual Studio Team System



27

Microsoft

In Zukunft: Integration von Tools für MOTION



28

Microsoft

Indigo

Einheitliches Programmiermodell zur Implementierung service-orientierter Architekturen

Einheitliches Programmiermodell

- ❖ Vereinheitlicht die heute in .NET existierenden Stacks zur Programmierung von verteilten Anwendungen.
- ❖ Für Applikationen die in lokalen Netzen als auch internet-weit verteilt sind.

Interoperabilität & Integration

- ❖ Interoperabilität mit Applikationen auf anderen Plattformen auf der Basis der WS*-Spezifikationen.
- ❖ Interoperabilität mit den heute in .NET existierenden Stacks für verteilte Anwendungen.

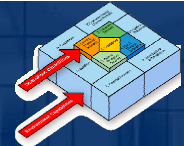
Vereinfachung von service-orientierter Programmierung

- ❖ Service-orientiertes Programmiermodell
- ❖ Unterstützt die vier Grundprinzipien für Services (Klar definierte Grenze, Autonom, ...)
- ❖ Verbesserung der Entwicklerproduktivität

29

Microsoft

Zusammenfassung



- ❖ SOA ist keine Revolution sondern eine Evolution und nutzt bewährte Ansätze und Methoden der Softwaretechnologie.
- ❖ SOA verspricht einen Return-on-Invest durch Kostenreduzierung und erhöhte Flexibilität bei der Implementierung von Geschäftsprozessen.
- ❖ Mit einem „Business Capability Model“ lassen sich die stabilen Elemente von Geschäftsprozessen strukturiert beschreiben und dann auf „Services“ abbilden.
- ❖ Die Etablierung einer SOA in einem Unternehmen ist ein mittel- bis langfristiger Prozess, der vom Management unterstützt werden muss.
- ❖ Microsoft stellt mit BizTalk Server 2004 und dem Information Bridge Framework eine solide technologische Basis zur Realisierung einer SOA zur Verfügung.

30

Microsoft

Zusätzliche Informationen

- ❖ David Sprott, Lawrence Wilkes, Understanding Service-Oriented Architecture, Microsoft Corporation, 2004
- ❖ Understanding BizTalk Server 2004, Microsoft Corporation, 2004, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bts2004is/htm/understanding_abstract_syfs.asp
- ❖ Understanding Microsoft Integration Technologies, A Guide to Choosing a Solution, Microsoft Corporation, 2005
- ❖ David Pallmann, Programming Indigo: The Programming Model, Microsoft Corporation, 2005, <http://msdn.microsoft.com/longhorn/default.aspx?pull=/library/en-us/dnlong/html/progindigo3.asp>
- ❖ Enterprise Solution Patterns Using Microsoft .NET Version 2.0, Microsoft Corporation, 2003, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/Esp.asp>
- ❖ DSL Tools Homepage <http://lab.msdn.microsoft.com/teamsystem/workshop/dsltools/>
- ❖ Software Factories <http://msdn.microsoft.com/architecture/overview/softwarefactories/>
- ❖ Eric Newcomer, Greg Lomow, Understanding SOA with web services, Addison-Wesley 2004, ISBN 0-321-18086-0

Fragen
???



Web Services

Ideale Plattform oder Mittel zum Zweck?

sd&m AG
software design & management
Carl-Wery-Str. 42
81739 München
Telefon 089 63812-0
www.sdm.de

A Company of



Dr. Harald Haller
TU München, 21. Juli 2005



Agenda



Agenda

■ Definition Web Services

Thesen zu Web Services

Stärken und Schwächen von Web Services

Fazit



Die Technologie der Web Services basiert auf vom W3C definierten Standards

Definition: Web Services (allgemein)

- **W3C:** A Web service is a
 - **software application**
 - **identified by a URI**
 - whose **interfaces** and binding are capable of being
 - **defined and discovered by XML artifacts** and
 - **supports direct interactions** with other software applications
 - **using XML based messages**
 - **via internet-based protocols**
- **Florescu:**
An autonomous piece of software uniquely identified by a URI, able to communicate with peer Web services via messages, using WWW specific standards and protocols (e.g. HTTP, XML, SOAP)

sd&m AG, 21.07.2005, Seite 3

Die wichtigsten Standards neben XML sind SOAP, WSDL und UDDI



Weitere Definitionen

- **SOAP (Simple Object Access Protocol)**
ist ein auf XML basierendes Protokoll für den Informationsaustausch in einer dezentralen und verteilten Landschaft
- **WSDL (Web Services Description Language)**
ist eine auf XML basierende Sprache zur Beschreibung der Web Services inkl. der Zugriffe
- **UDDI (Universal Description, Discovery and Integration)**
standardisiert den Aufbau von Verzeichnissen und fungiert damit als eine Art „Gelbe Seiten“ für Web Services

sd&m AG, 21.07.2005, Seite 4



Agenda

Definition Web Services

■ **Thesen zu Web Services**

Stärken und Schwächen von Web Services

Fazit

sd&m AG, 21.07.2005, Seite 5



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

6 Thesen zu Web Services

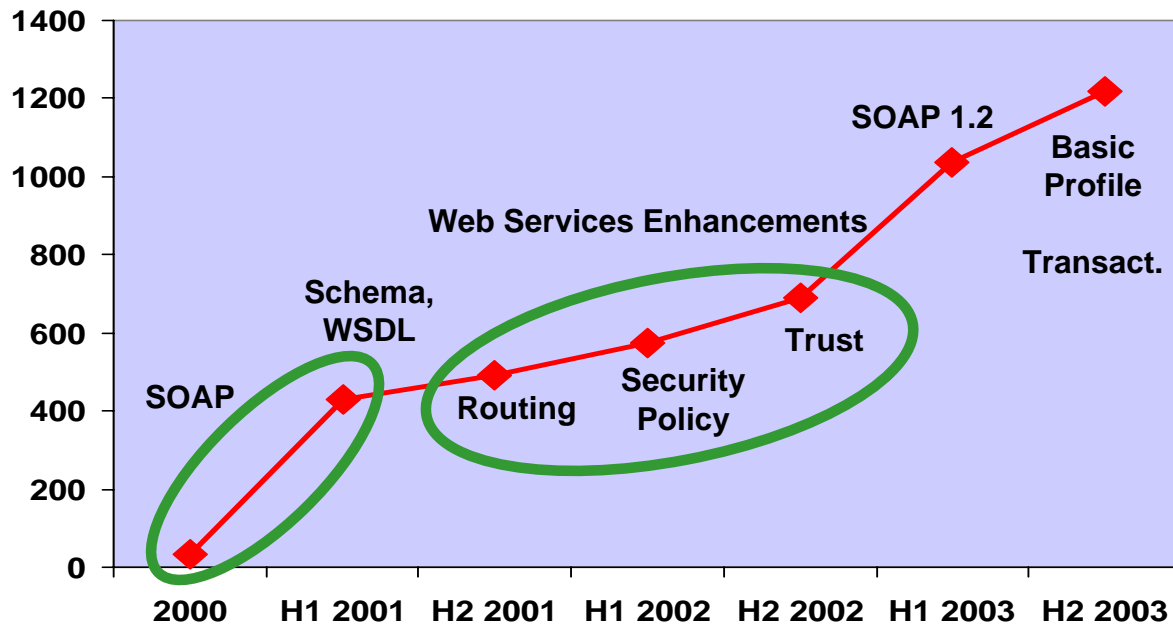
1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.

sd&m AG, 21.07.2005, Seite 6



Die Spezifikationen für die Basistechnologien waren bereits im ersten Halbjahr 2001 vorhanden. Trotzdem entstehen immer noch umfangreiche Dokumente

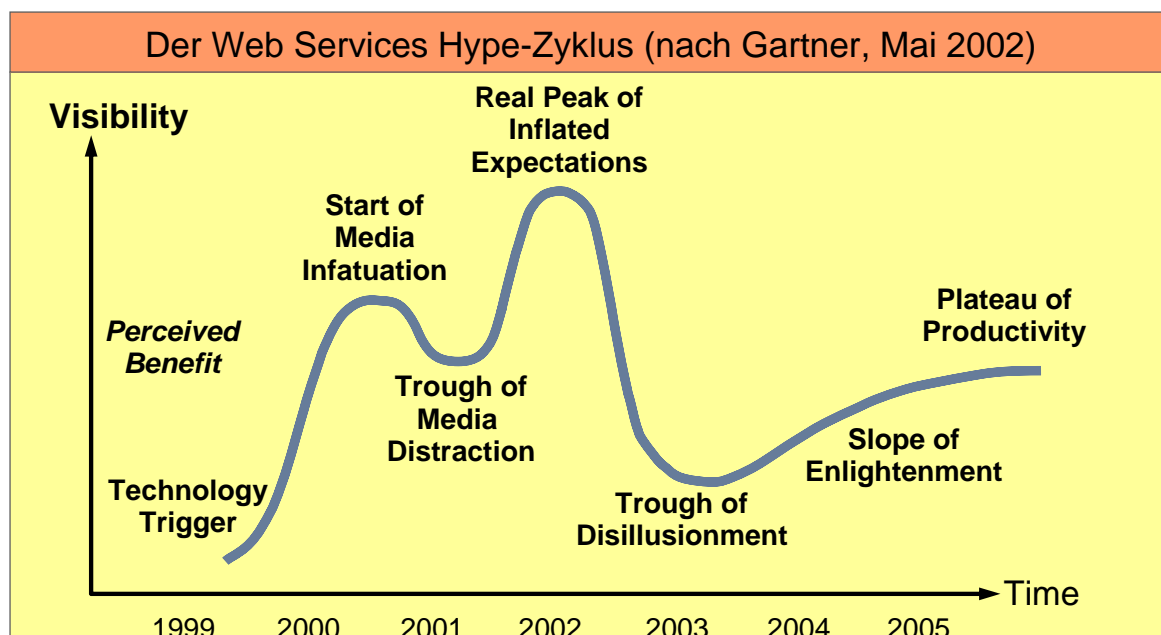
Gedruckte Seiten für Spezifikationen



sd&m AG, 21.07.2005, Seite 7



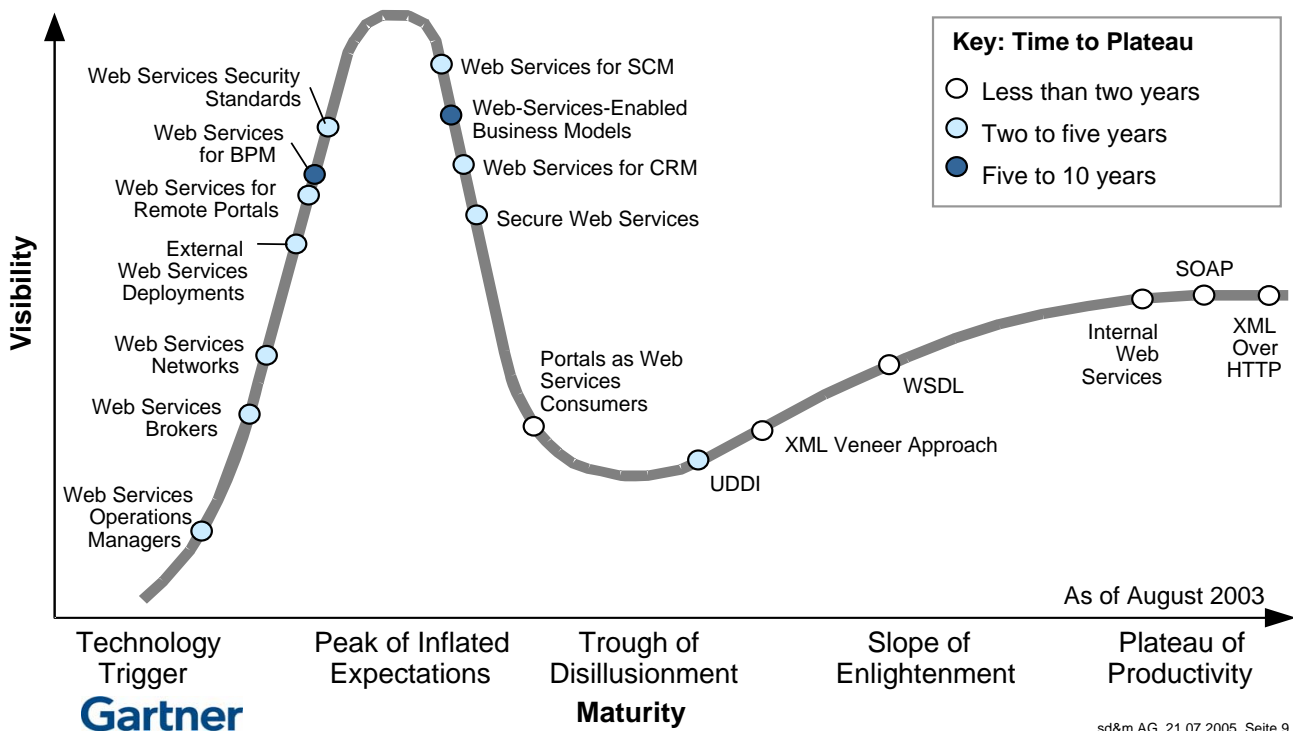
Nach Gartner erreichen Web Services jetzt das „Plateau of Productivity“



sd&m AG, 21.07.2005, Seite 8



Die Basis-Technologien sind nahezu ausgereift. Trotzdem gibt es noch viele Themen, die für ihre Reife noch Jahre benötigen



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

6 Thesen zu Web Services

1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.
2. Web Services sind vielfältig und flexibel einsetzbar.



Die Basistechnologie XML besitzt viele Vorteile, die sich direkt auf Web Services übertragen. Der Hauptnachteil ist eine gewisse „Geschwätzigkeit“ (Overhead)

Vorteile von XML

- Einfach und flexibel (unbekannte Schemata), unregelmäßige Daten: z.B. fehlende Felder
- Lesbar für Menschen und Maschinen
- Selbstbeschreibend
- Portabel, plattformunabhängig, universell (z.B. China)
- XML ist nicht nur ein Datenformat
- „Standard“ des W3C
- XML ist etabliert
- Grundlage für viele Tools und weitere Standards
 - Parser, Anfragesprachen, Datenbanken, Visualisierung

sd&m AG, 21.07.2005, Seite 11



Die Vorteile von XML übertragen sich direkt auf das Kommunikationsprotokoll SOAP

Weitere zentrale Technologien

- SOAP (**S**imple **O**bject **A**ccess **P**rotocol)
 - Kommunikationsprotokoll für die Anwendungen
 - Basierend auf XML
 - Plattform-, Programmiersprachenunabhängig
 - Einfach und erweiterbar
 - Viele weitere Standards (Encryption, ...)
 - Flexibel einsetzbar (auch im Internet und mit Firewalls)

sd&m AG, 21.07.2005, Seite 12



Weitere Technologien dienen der Beschreibung und dem Finden von Services

Weitere zentrale Technologien

- **WSDL (Web Service Description Language)**
 - Beschreibt die (statische) Schnittstelle eines Dienstes in XML
 - Keine Beschreibung von
 - dynamischen Aspekte (Choreographie),
 - SLA, Transaktionen, Kosten (WS Policy)
 - Eigenschaften für Optimierung (Seiteneffekte, Kompensierbarkeit, ...)
- **UDDI (Universal Description Discovery Integration)**
 - „Gelbe Seiten“ fürs Web (speichert die WSDL-Dokumente)
 - Keine Zusicherungen über angebotenen Dienst
 - Kein erweiterbares Datenmodell

sd&m AG, 21.07.2005, Seite 13



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

6 Thesen zu Web Services

1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.
2. Web Services sind vielfältig und flexibel einsetzbar.
3. Web Services sind eine mögliche Integrationstechnik für SOA. Sie sind insbesondere bei der Integration verschiedener Anwendungslandschaften sinnvoll einzusetzen.

sd&m AG, 21.07.2005, Seite 14



Web Services sind etabliert

Aus einer Umfrage zum Thema Web Services und SOA

- Die Vorteile von Web Services und ihre Möglichkeiten werden von den Managern und Entscheidern richtig eingeschätzt
- 35 % der Unternehmen
 - planen den Aufbau einer Web Service-Infrastruktur
 - sind gerade beim Bau oder
 - haben diese im Einsatz
- 18 % der befragten Unternehmen sind „SOA-ready“
- 24 % haben „Punkt zu Punkt“-Anwendungen mit Web Services
- 29 % sind in der Entwicklung der Web Services

Quelle: "From Web Services to SOA and Everything in Between: The Journey Begins",
11. Mai 2005, <http://www.webservices.org/index.php/ws/content/view/full/63404>

sd&m AG, 21.07.2005, Seite 15



Bereits mehr als zwei Drittel der befragten Unternehmen haben Web Services im Einsatz

Anzahl der Web Services aktuell im Einsatz	
Keine	30 %
1 – 5	34 %
6 – 20	18 %
21 – 50	18 %

Quelle: "From Web Services to SOA and Everything in Between: The Journey Begins",
11. Mai 2005, <http://www.webservices.org/index.php/ws/content/view/full/63404>

sd&m AG, 21.07.2005, Seite 16



Die Integration eigener interner Systeme ist der Hauptgrund für den Einsatz von Web Services

Warum werden Web Services implementiert?	
Integration verschiedener Systeme	35.6 %
Erweiterung des Online-Geschäfts	12.6 %
Wichtiger Kunde fordert Web Services-Funktionalität	11.4 %
Entwickler möchten Web Services testen	8.5 %
Erhöhung der Entwicklerproduktivität	6.8 %
Ersetzung oder Verbesserung von EDI (Electronic Data Interchange)-Services und -Systemen	3.5 %
Ein verwendetes Softwareprodukt erfordert Web Services	2.6 %
Ein wichtiger Lieferant bietet Web Services-Schnittstelle(n)	1.1 %

Quelle: "From Web Services to SOA and Everything in Between: The Journey Begins",
11. Mai 2005, <http://www.webservices.org/index.php/ws/content/view/full/63404>

sd&m AG, 21.07.2005, Seite 17



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

6 Thesen zu Web Services

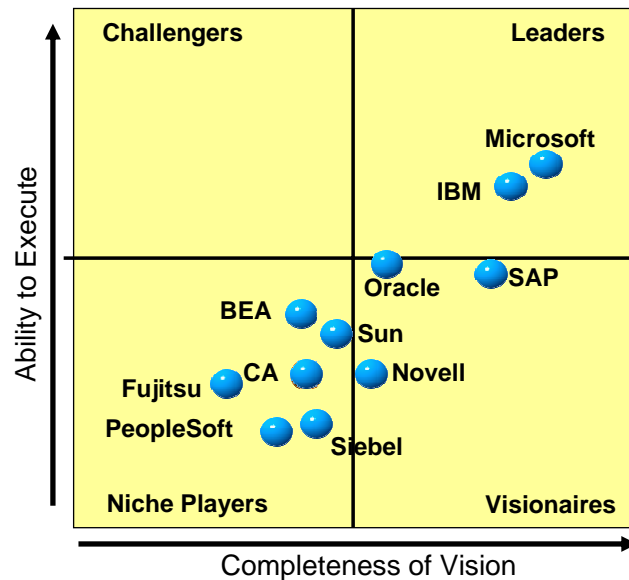
1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.
2. Web Services sind vielfältig und flexibel einsetzbar.
3. Web Services sind eine mögliche Integrationstechnik für SOA. Sie sind insbesondere bei der Integration verschiedener Anwendungslandschaften sinnvoll einzusetzen.
4. Web Services werden von den großen Softwareherstellern unterstützt.

sd&m AG, 21.07.2005, Seite 18



Die großen Software- und Hardware-Produkthersteller (insbesondere Microsoft, IBM, SAP) setzen auf Web Services

Gartner Magic Quadrant: Major vendor Web services platform influence



Source: Gartner Research, September 2004

sd&m AG, 21.07.2005, Seite 19



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

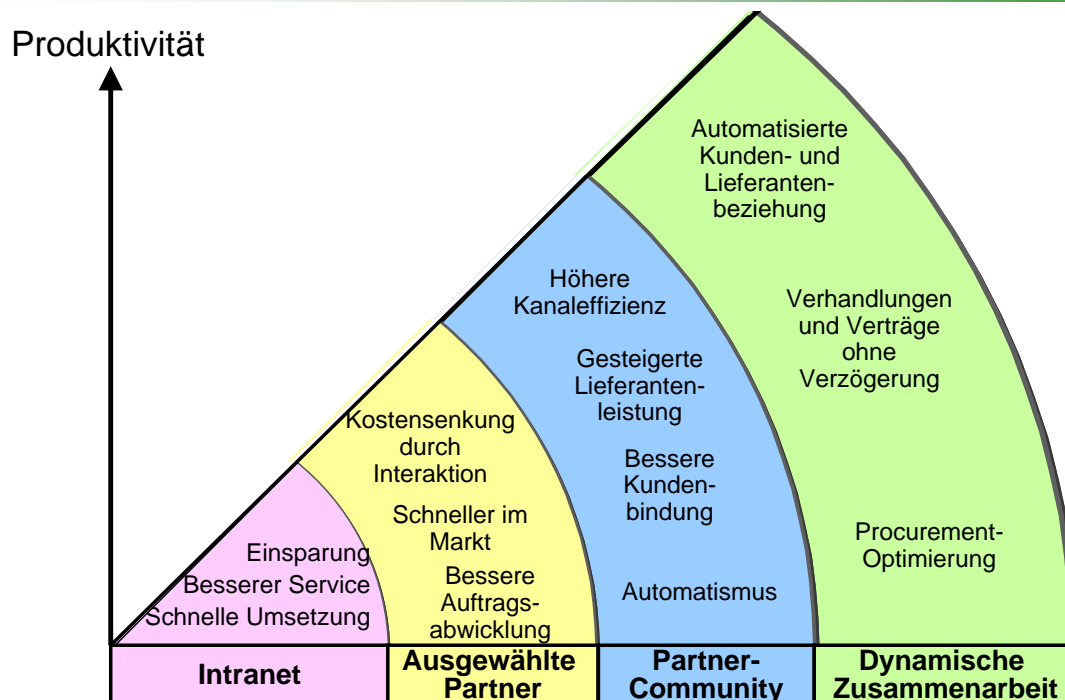
6 Thesen zu Web Services

1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.
2. Web Services sind vielfältig und flexibel einsetzbar.
3. Web Services sind eine mögliche Integrationstechnik für SOA. Sie sind insbesondere bei der Integration verschiedener Anwendungslandschaften sinnvoll einzusetzen.
4. Web Services werden von den großen Softwareherstellern unterstützt.
5. Web Services entfalten Ihr Potenzial erst richtig außerhalb von Unternehmensgrenzen.

sd&m AG, 21.07.2005, Seite 20



Web Services entfalten ihre Möglichkeiten vor allem bei Diensten, die außerhalb des Unternehmens angeboten werden



sd&m AG, 21.07.2005, Seite 21



Es gibt einige wichtige Aspekte, um die Bedeutung und den Nutzen von Web Services zu bewerten

6 Thesen zu Web Services

1. Web Services bestehen aus einer Fülle von Einzelspezifikationen, die teilweise noch nicht endgültig finalisiert sind.
2. Web Services sind vielfältig und flexibel einsetzbar.
3. Web Services sind eine mögliche Integrationstechnik für SOA. Sie sind insbesondere bei der Integration verschiedener Anwendungslandschaften sinnvoll einzusetzen.
4. Web Services werden von den großen Softwareherstellern unterstützt.
5. Web Services entfalten Ihr Potenzial erst richtig außerhalb von Unternehmensgrenzen.
6. Für die Entwicklung einer SOA gibt es Alternativen zu Web Services, die in Einzelfällen Web Services überlegen sein können.

sd&m AG, 21.07.2005, Seite 22

Gegenüber anderen Web Services-Technologien hat sich SOAP durchgesetzt



Alternativansätze zu Web Services

- SOAP-ähnliche Ansätze für Web Services, wie
 - XML-RPC
 - Quasi Vorläufer von SOAP
 - Es gab einen engen Austausch der Entwickler von SOAP und XML-RPC bei Microsoft
 - Das Whitepaper zu XML-RPC (Veröffentlichung im Jahre 1998) ist nahezu unverändert geblieben
 - Die Grundspezifikation des Protokolls mit den Formaten für Request, Response und Faults umfasst gerade einmal sieben Seiten
 - REST-Frameworkspielen praktisch keine Rolle
- SOAP hat sich gegenüber diesen durchgesetzt, vor allem durch den Rückenwind von Apache, Microsoft u.a.

sd&m AG, 21.07.2005, Seite 23

Thesen zu Web Services

Die zahlreichen Alternativen zu Web Services bieten Optimierungsmöglichkeiten auf Kosten von Flexibilität und Standardkonformität



Alternativen zu Web Services

- Für den Aufbau einer SOA sind andere Protokolle und Middleware-Produkte geeignet.
- Dazu gehören
 - Corba
 - RMI
 - .NET Remoting
 - Transaktionsmanager
 - HTTP
 - RPC
 - COM, DCOM, COM+
 - ...

Allgemeine Bewertung

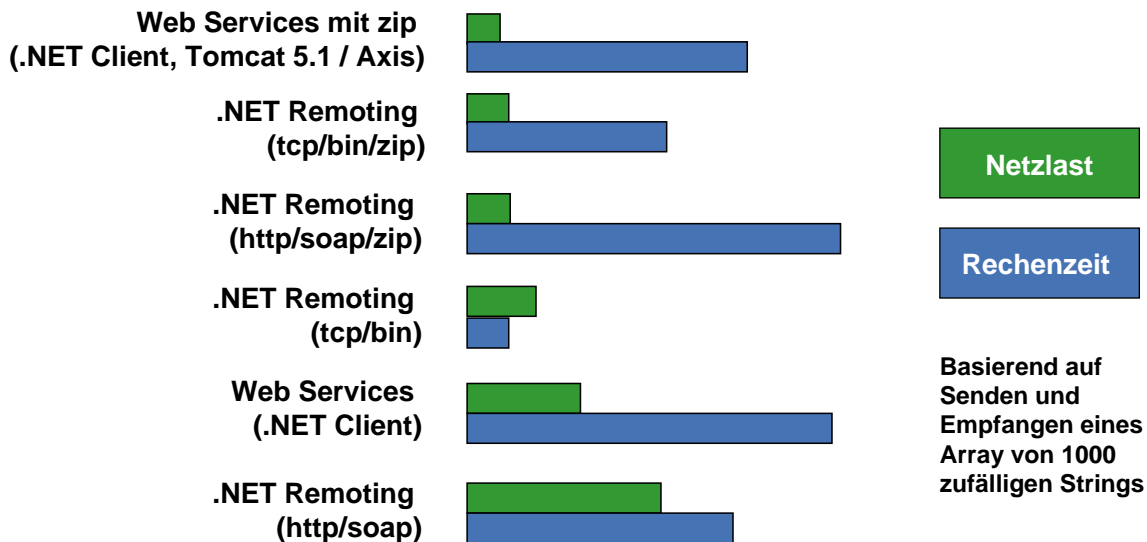
- Mit spezifischen Protokollen sind Optimierungen möglich bzgl.
 - geringere Datenredundanz (Performanz)
 - schnellere Aufrufe
 - erweiterte Funktionalität
- Man verliert (je nach Einzelfall)
 - Standardkonformität
 - Plattformunabhängigkeit
 - Flexibilität

sd&m AG, 21.07.2005, Seite 24



Die Antwortzeiten und die Netzlast lassen sich im Vergleich zu puren Web Services verbessern

Antwortzeiten und Netzlast bei verschiedenen Kommunikationskanälen



sd&m AG, 21.07.2005, Seite 25

Bereits Microsoft bietet eine Fülle verschiedener Varianten für verteilte Anwendungen



Microsoft-Technologien für verteilte Anwendungen unter .NET

- Verschiedene Technologien für verteilte Anwendungen mit .NET/Windows
 - Web Services
 - Enterprise Services (COM+)
 - Remoting
 - Message Queuing (MSMQ)
 - Low-Level TCP Sockets
 - UDP Datagrams
 - Named Pipes
 - ...
 - Zahlreiche Toolkits von Drittanbietern
 - Remoting.CORBA
 - XMLRPC.Net
 - Peer-To-Peer-Implementierungen
 - ...

sd&m AG, 21.07.2005, Seite 26



Agenda

Definition Web Services

Thesen zu Web Services

■ **Stärken und Schwächen von Web Services**

Fazit

Stärken und Schwächen von Web Services

Web Services sind als Standard im Markt akzeptiert. Wichtig ist es, möglichst keine proprietären Erweiterungen zuzulassen



Web Services	Stärken	<ul style="list-style-type: none"> + Einfach anwendbar, selbstbeschreibend + Flexibel einsetzbar und erweiterbar + Plattform- und Produktunabhängig + Als Standard akzeptiert + Umfangreiche Unterstützung für Einsatz durch Produkthersteller + In Praxis bewährt 	Schwächen	<ul style="list-style-type: none"> – „Geschwätzig“ – Ineffizient bzgl. Datenvolumen – Viele einzelne und umfangreiche Spezifikationen, teilweise nicht finalisiert – Existenz proprietärer Erweiterungen
	Chancen	<ul style="list-style-type: none"> ! Weitere Verbreitung im Markt ! Zunehmende Entwicklung von Services im Internet ! Sinnvolle Erweiterung und Finalisierung der Spezifikationen 	Risiken	<ul style="list-style-type: none"> ? Anforderungen führen zu proprietären Erweiterungen ? Key Player im Markt versuchen eigene Wege zu gehen ? Vorurteile bzgl. Zuverlässigkeit und Sicherheit bleiben bestehen



Agenda

Definition Web Services
Thesen zu Web Services
Stärken und Schwächen von Web Services

■ **Fazit**

sd&m AG, 21.07.2005, Seite 29



Web Services sind nicht immer die „ideale Plattform“, aber ein sehr gutes „Mittel zum Zweck“

Hauptargumente für die Bedeutung von Web Services

- Web Services sind einfach anwendbar und flexibel einsetzbar.
- In Einzelfällen bieten andere Technologien mehr Optimierungspotenzial
- Web Services sind als Standard akzeptiert und haben sich in der Praxis bewährt
- Web Services werden durch Produkthersteller sehr gut unterstützt
- Deshalb werden Web Services auch die beherrschende Technologie für SOA (zumindest auf absehbare Zeit) bleiben

sd&m AG, 21.07.2005, Seite 30



“It is not the strongest of the species that survive,
nor the most intelligent,
but the ones most responsive to change.”
(Charles Darwin)

sd&m AG, 21.07.2005, Seite 31

Kontakt

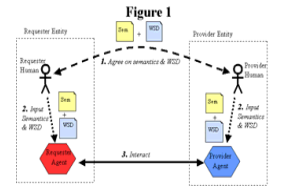


Dr. Harald Haller
Tel. 089 63812-431
haller@sdm.de

sd&m AG
Carl-Wery-Str. 42
81739 München
Telefon 089 63812-0
Fax 089 63812-330
info@sdm.de

SOA – Von der Vision zur Realität

Gino Popescu (HVB Systems)



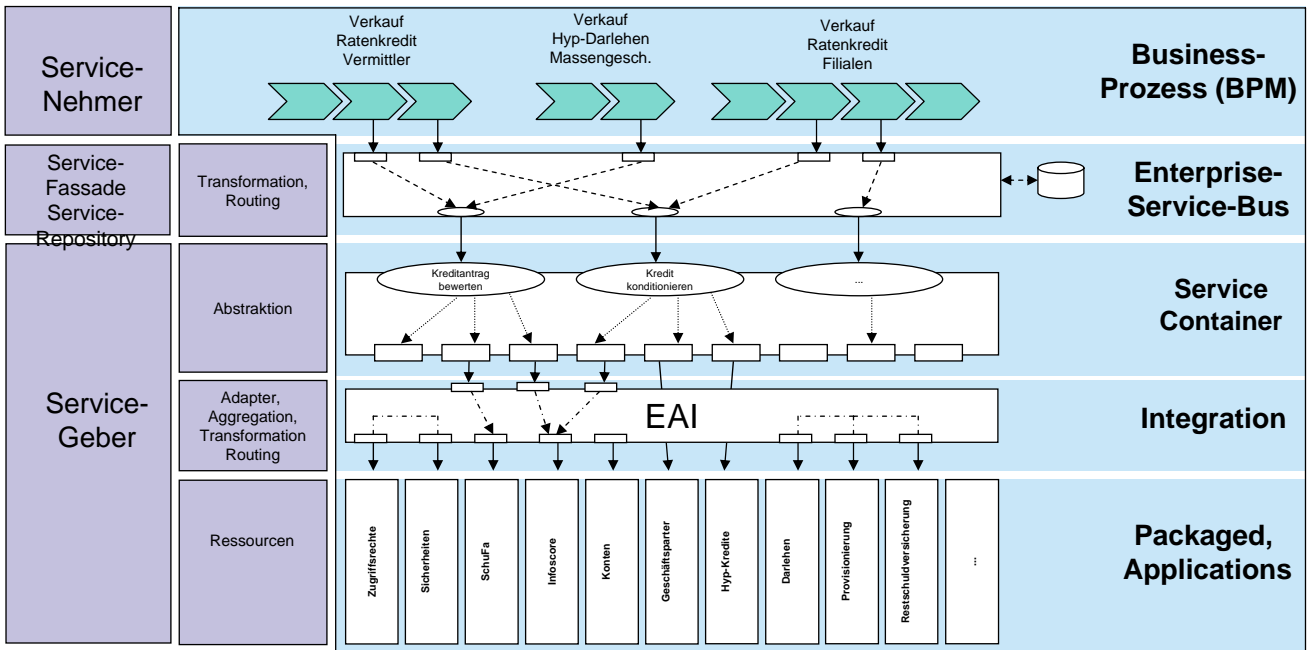
Ein Mitglied der HVB Group

Agenda

- Vision HVB bzgl. SOA
- Nutzenpotentiale von SOA
- typische Fragestellungen bei der Umsetzung einer SOA
- Lessons Learned



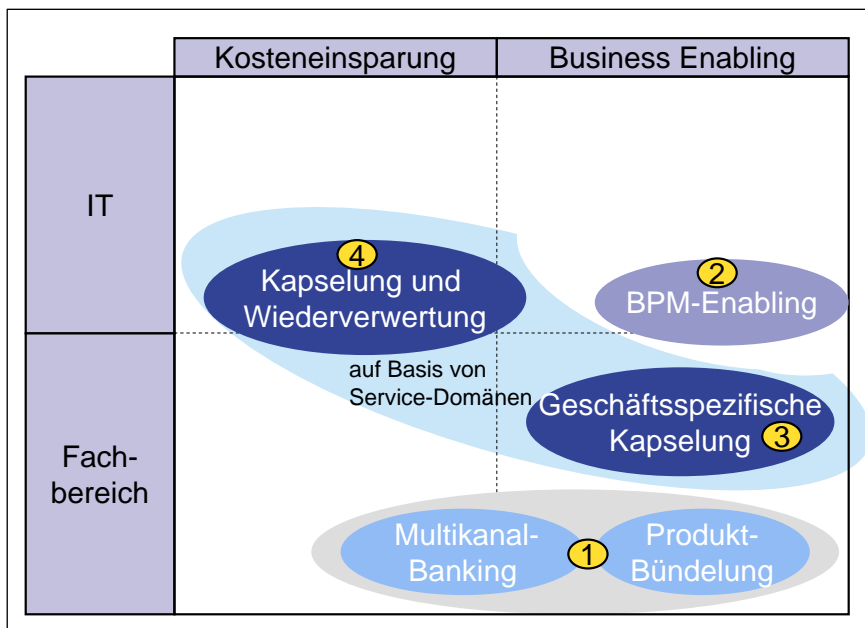
SOA – Services bilden fachliche Teilprozesse ab



SOA-Von der Vision zur Realität; Gino Popescu
3

Ein Mitglied der HVB Group

Optimierungs bzw. Nutzenpotentiale durch SOA



- 1. Produktbündelung/ Multikanalbanking**
Wiederverwendung über Kanäle und Produkte
- 2. BPM-Enabling**
Direkte Integration innerhalb der Geschäftsprozesse
- 3. Geschäftsspezifische Kapselung**
Bankfachliche Kapselung von Teilbereichen erhöht die Flexibilität
- 4. Kapselung und Wiederverwertung**
Funktionale Kapselung auf Basis technischer Standards erhöht die Wiederverwertung und garantiert mehr Stabilität

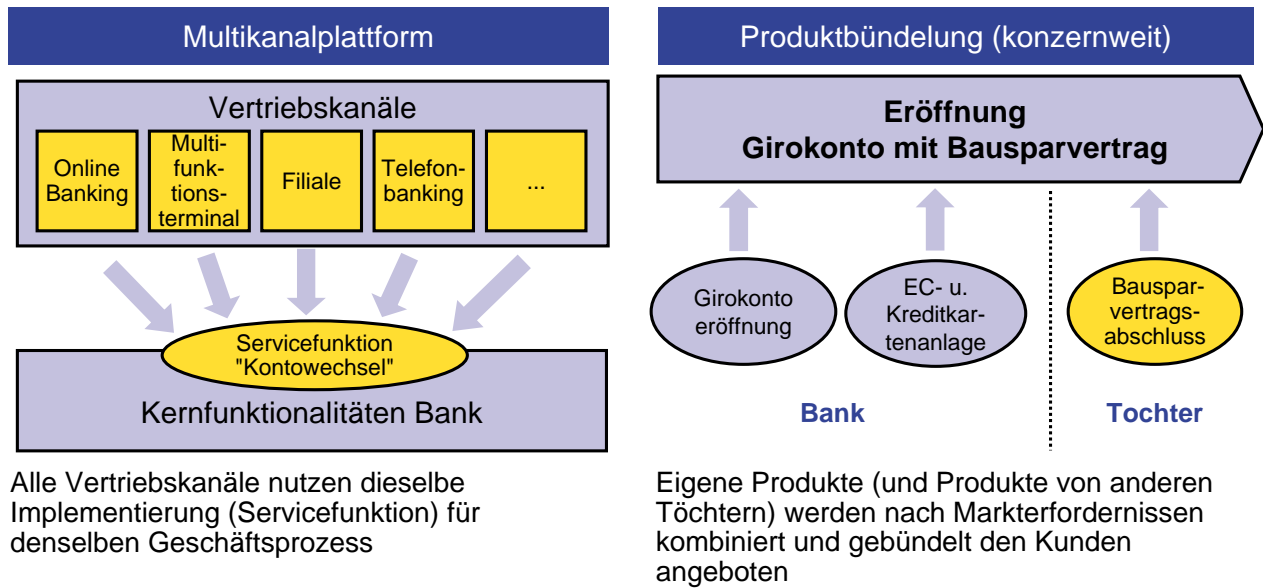


SOA-Von der Vision zur Realität; Gino Popescu
4

Ein Mitglied der HVB Group

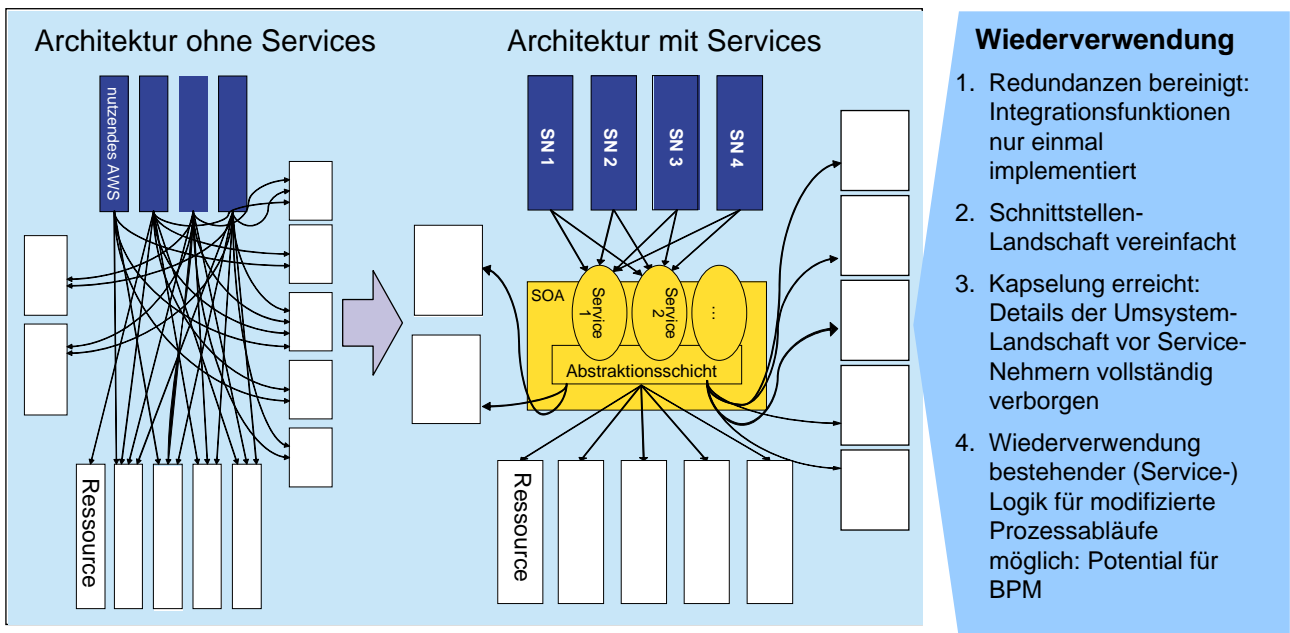
SOA Architektur und Nutzen-Potentiale in der Bank

SOA Chance 1: Multikanal und Produktbündelung



SOA Architektur und Nutzen-Potentiale in der Bank

SOA-Chance 4: Wiederverwendung und Vereinfachung



notwendige Aktivitäten zur Einführung

- strategische Entscheidung des Unternehmens eine SOA in größerem Stiel einzuführen
- Evolutionäre Einführung über strategische Projekte über ein projektübergreifendes "Team" um Synergien zu erkennen und zu heben
- Entwicklung eines firmenspezifischen Governance-Modells (Organisationsmodell, Service-Zuordnung, Leistungsverrechnung, Verantwortlichkeiten, Service-Level-Agreement, etc.)
- Bereitstellung der technischen Infrastruktur (ESB incl. Repository, Tools, etc.)
- Entwicklung eines Vorgehensmodells für SOA-Projekte (zusätzliche Phasen für Service-Erkennung und -Spezifikation)
- Vermarktung der Idee im Unternehmen, um sie dort zu verankern

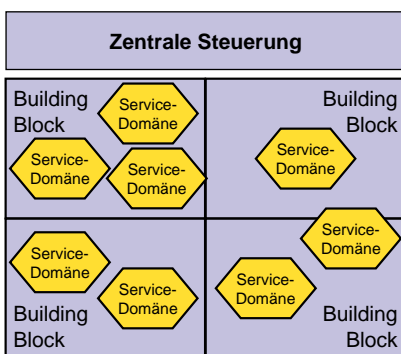


SOA-Von der Vision zur Realität; Gino Popescu
7

Ein Mitglied der HVB Group

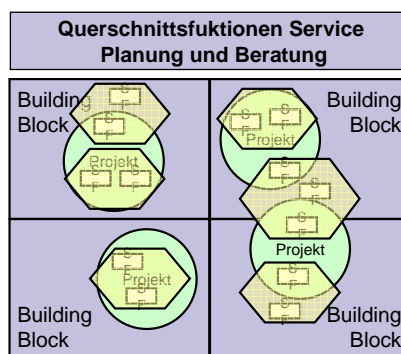
Einführungsstrategien und Governance

Top-Down



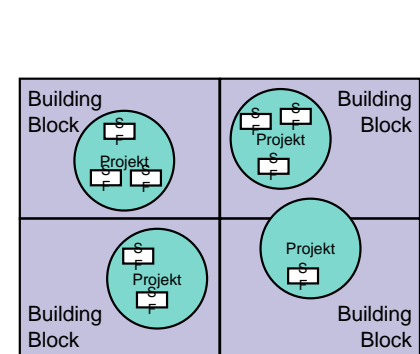
- Vorteile:
- + Konsistentes Modell durch Gesamtsicht
- Nachteile:
- Hohe Initialkosten
 - Extrem hohe Komplexität

Evolutionär-Gesteuert



- Vorteile:
- + Services entstehen im Projektbetrieb
 - + Gesamtsicht auf Prozesse und Service-Domänen bleibt erhalten
 - + Keine Inseln von Servicefunktionen
- Risiko:
- mangelnde Akzeptanz der Steuerung

Bottom-Up



- Vorteile:
- + Services entstehen im Projektbetrieb
- Nachteile:
- "unvollständige" Services, da Projektscope i.A. nicht auf Servicedomänen matcht
 - Verlust der Idealsicht wegen zu enger Kopplung an bestehende Prozesse oder Systeme.

Servicefunktion
Building-Block=fachliche Einheit



SOA-Von der Vision zur Realität; Gino Popescu
8

Ein Mitglied der HVB Group

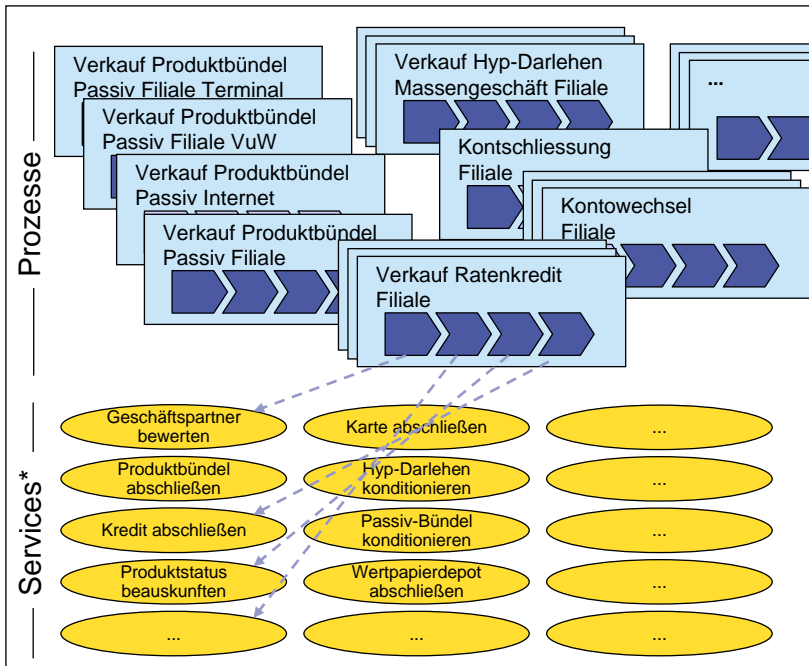
Lessons Learned

- Die nutzenbringende Einführung einer SOA bedingt einen ganzheitlichen Ansatz, der sich über alle Strukturen und Methodiken des Unternehmens erstreckt. Eine SOA kann daher nicht gekauft sondern muß für das Unternehmen zugeschnitten werden.
- Die richtige Granularität von Services und der Abstraktionsgrad sind nur schwer zu bestimmen. Hier ist ein Hybrid-Modell zu bevorzugen, bei dem die direkten Projekt-Anforderungen gegen die Anforderungen des Geschäftsbereich gespiegelt werden und ständig gegeneinander abgestimmt werden müssen.
- Das Governance-Modell für die Einführung einer SOA impliziert eine Abkehr vom klassischen Governance-Modell für die normale Anwendungsentwicklung
- Fachliche Service können technisch relativ einfach umgesetzt werden, auch wenn viele Fragestellungen noch nicht restlos geklärt sind (z.B. Webservices - Security und Transaktionshandling). WebServices-Standards (z.B. Protokolle, Interface-Definitionen) und EAI-Methodiken (z.B. Transformation, Messaging, Aggregation) sind Techniken die dabei unterstützen können.

Backup

SOA Architektur und Nutzen-Potentiale in der Bank

SOA Chance 2: BPM-Enabling



■ Ziel:

Hochwertige Unterstützung von Business Prozessen:

- wenige "Expertensprünge"
- möglichst weitgehende Automatisierung

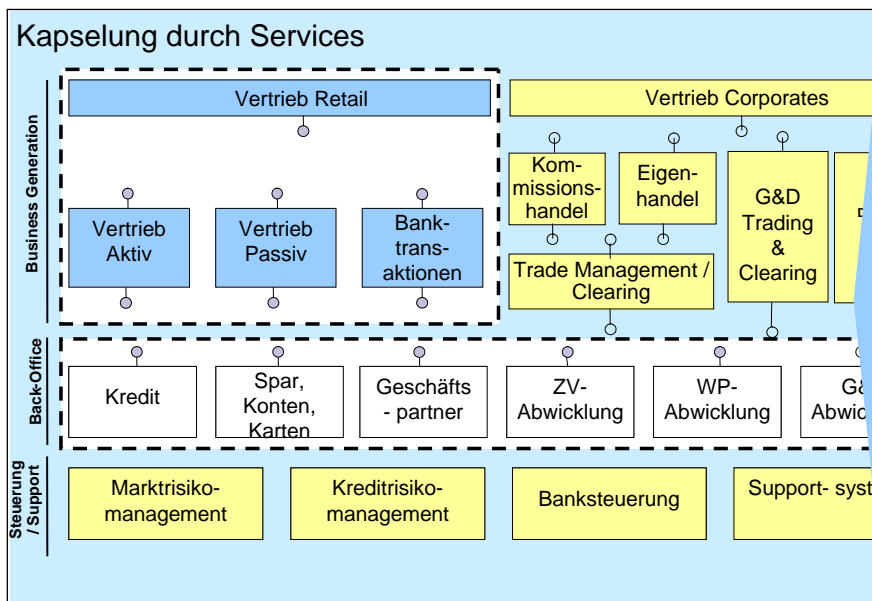
■ Unterstützung durch SOA:

- Prozessteile durch Services implementiert und einfach in Prozessen nutzbar
- Prozessunterstützung daher einfach realisierbar (-> Services sind "plug-and-play")



SOA Architektur und Nutzen-Potentiale in der Bank

SOA Chance 3: Flexibilität durch Kapselung in Servicedomänen



Durch Kapselung erzielbare Potentiale:

- **Flexibles Sourcing von IT-Lösungen** durch Erleichterung Einführung und Austausch von Packages
- **Insourcing oder Merger:** Erleichterung Unterstützung von Drittbanken durch eigene, über Services nutzbare Systeme
- **Outsourcing:** Erleichterung Nutzung Serviceleistungen von Drittanbietern durch klare Abgrenzungen



SOA

Zwischen Anspruch und Wirklichkeit

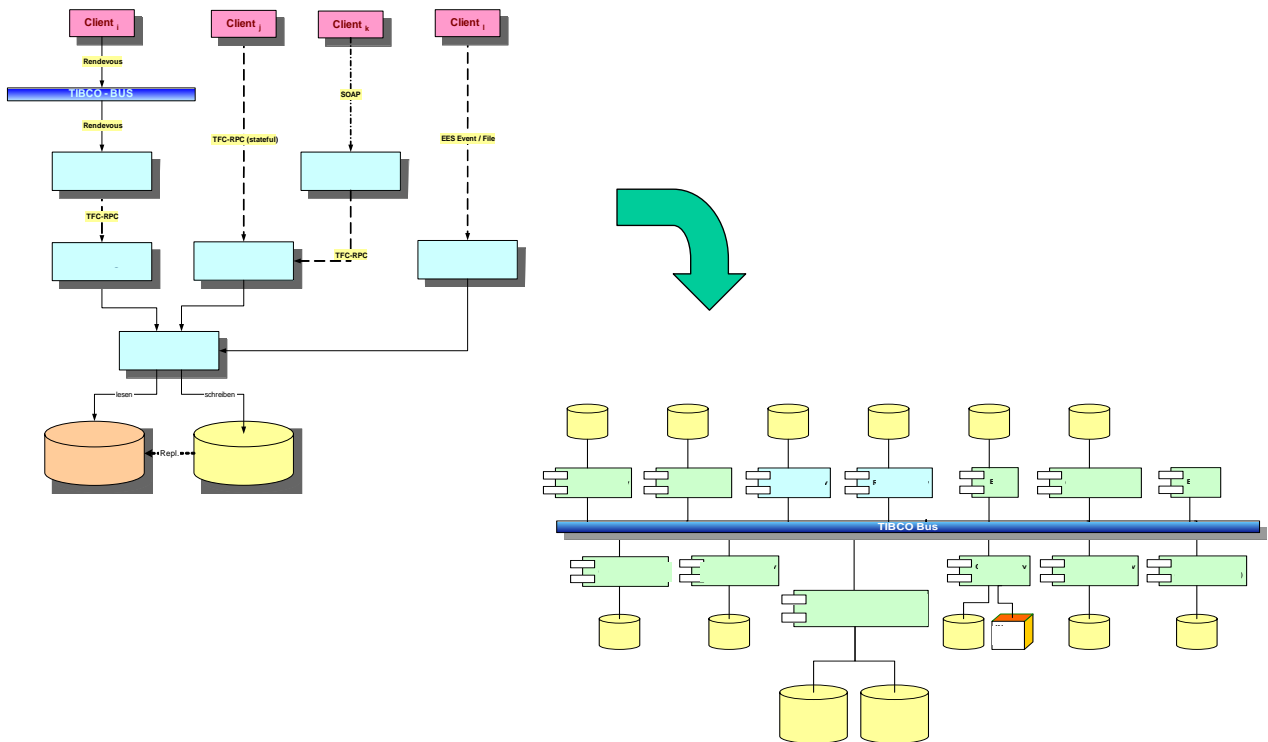
Nicolai M. Josuttis
IT-communication.com

©2005 by IT-communication.com

Erfahrungs-Kontext

- **Bereich Telekommunikation**
- **Verwaltung von zig-Millionen Kundendaten**
- **Heterogenes Umfeld von etlichen Anwendungen**
- **Service-Umfeld betrifft mehrere Hundert Entwickler**
- **Modellierung der Services mit Rational-Rose (MDA)**

„Schöne neue Welt“



©2005 by IT-communication.com

Auswahl einiger wichtiger Fragestellungen I

- **Granularität der Services?**
 - von Zugriff auf Fremdsysteme
 - bis Entkopplung von GUI und Business-Logik
- **Organisatorische Flaschenhälse**
 - zentrales Repository oder bilaterale Abstimmungen?
- **Sicherheitsaspekte**
 - Stateful oder Stateless?
 - Wo werden Rechte auf Daten geprüft?

©2005 by IT-communication.com

Auswahl einiger wichtiger Fragestellungen II

- **Typisierung**
 - generisch?
 - klassische Signatur (Datentypen der Parameter)?
 - Wertebereiche (Aufzählungstypen für Werte)?
 - SLA-Aspekte (z.B. Maximallaufzeit)?
- **Stabilität und Flexibilität**
 - Wann werden Service-Schnittstellen festgelegt?
 - Wasserfall?
- **Versionierung**
 - Sonderbehandlung von Änderungen, die rückwärtskompatibel sind? Änderungen?

©2005 by IT-communication.com

SOA braucht Sorgfalt und Zeit

- **SOA ist eine technische Infrastruktur, die in keinsten Weise fachliche Fragen und Rollenaufteilungen löst**
- **Es ist fatal, Konzepte abzuschliessen, ohne deren Pragmatismus durch Realisierung zu prüfen**
- **Es dauert Jahre**
- **Die Komplexität und Abhängigkeiten können leicht steigen (es wird technisch leicht, fachlich Dinge komplex zu machen)**
- **Rose ist eine Katastrophe (MDA nicht)**

©2005 by IT-communication.com

Hypothesen

- **Es hat den Anschein, dass der gleiche Fehler gemacht wird, wie bei der Durchsetzung der objektorientierten Idee**
 - Damals schien es alle Probleme zu lösen, wenn man Daten und Operationen zu Objekten zusammenschließt und dann die Objekte miteinander kommunizieren lässt
 - Heute wissen wir, dass Objektorientierung nicht skaliert. Große Systeme bestehen aus Komponenten, Schichten, etc.
 - Einen Bus mit gleichberechtigten Services zu machen birgt das gleiche Problem.
 - Wenn jeder alles aufrufen kann, droht ein Chaos.

©2005 by IT-communication.com

Vielen Dank



Nicolai Josuttis

www.it-communication.com
josuttis@it-communication.com

Gaussstr. 29
D - 38106 Braunschweig

Tel.: +49 531 / 129 88 86
+49 700 / 5678 8888
+49 700 / JOSUTTIS

©2005 by IT-communication.com

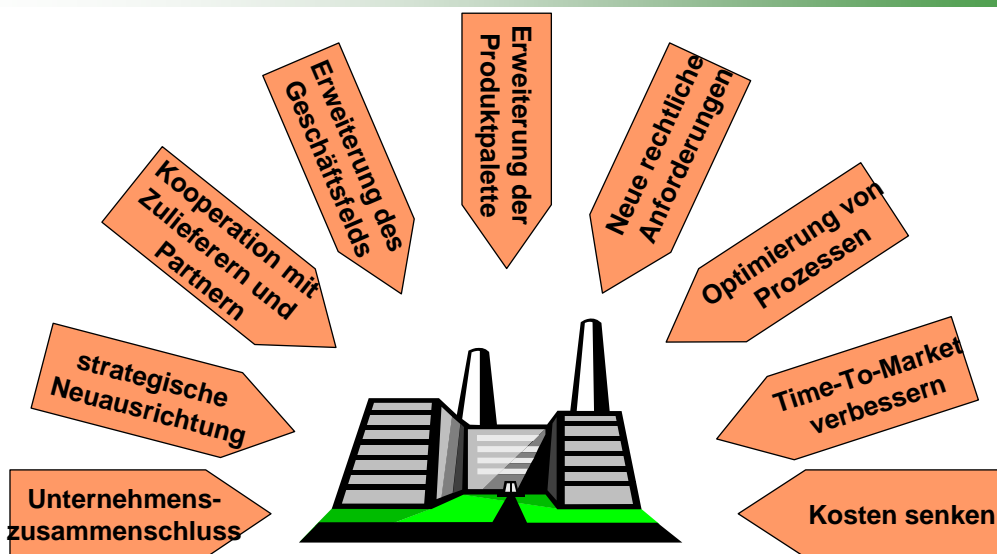


Wann liefert eine Serviceorientierte Architektur echten Nutzen?

Dr. Jan-Peter Richter
Hot Spots der Software Entwicklung 2005

Motivation

Die wirtschaftliche Agilität eines Unternehmens wird heute bestimmt durch die Flexibilität seiner IT



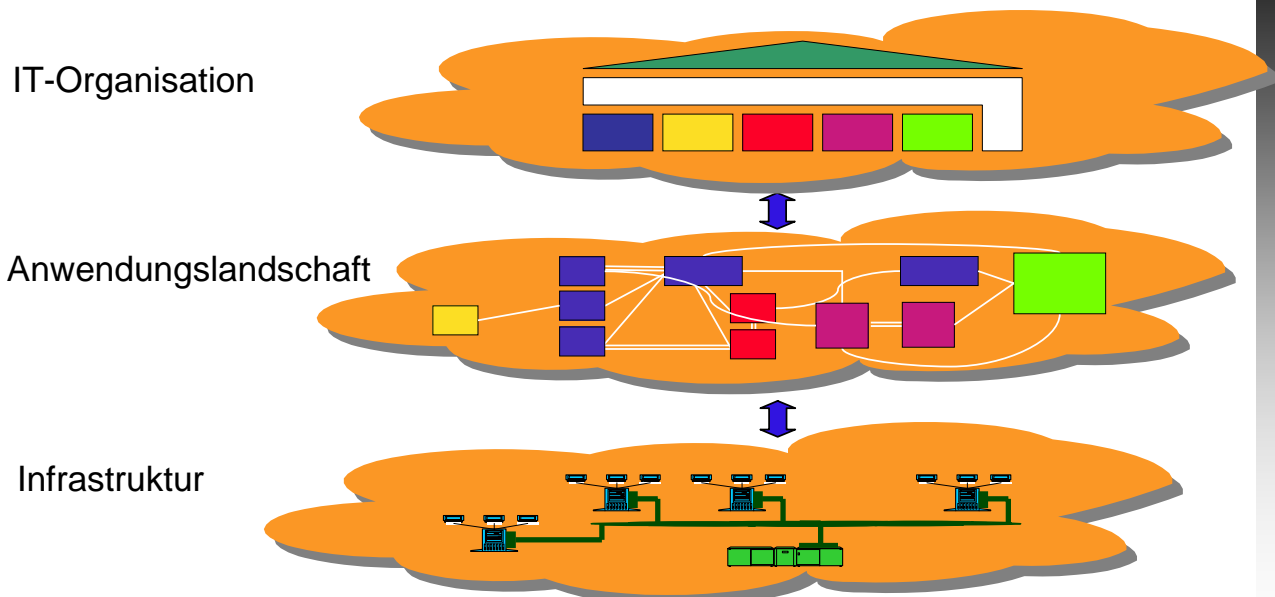
Die IT-Systeme müssen **flexibel**, **offen** und **effizient** sein, damit ein Unternehmen diese Herausforderungen bewältigen kann.

Die Serviceorientierte Architektur verspricht diese Flexibilität durch lose gekoppelte Services

- *Serviceorientierte Architektur* ist nur schwach definiert:
 - Betrifft die Anwendungslandschaft
 - Anwendungssysteme sind über *Services* lose gekoppelt
 - *Service* ist feste, definierte Leistung; Baustein größerer Verarbeitungsabläufe
 - abstrakte Sicht auf das anbietende Anwendungssystem
 - verbirgt Implementationsdetails
 - Definition hat den Charakter einer vertraglichen Übereinkunft
 - Aufruf erfolgt netztransparent über einheitlichen Mechanismus
- Beliebigkeit des Service-Begriffs erhöht die Gefahr des Scheiterns
- Grundgedanken der SOA sind im Software Engineering wohlbekannt

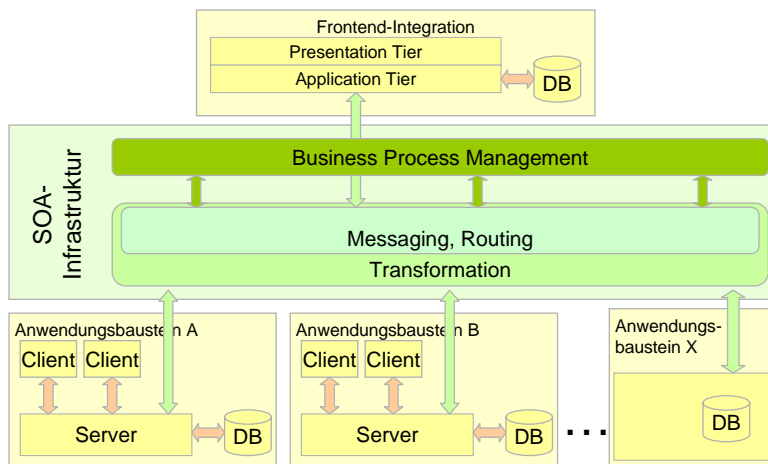
sd&m AG, 22.07.2005, Seite 3

Eine erfolgreiche SOA-Einführung betrifft alle Ebenen der Unternehmens-IT



sd&m AG, 22.07.2005, Seite 4

SOA benötigt eine Infrastruktur – sie ist aber nicht die Infrastruktur



- Einheitlicher Interaktionsmechanismus senkt TCO bei
 - Entwicklung
 - Wartung
 - Betrieb

- Anforderungen an die SOA-Infrastruktur sind unternehmensabhängig
- Es existiert kein Standard, keine „richtige“ Lösung
- Auswahl und Aufbau der Infrastruktur ist individuell und sorgfältig durchzuführen

sd&m AG, 22.07.2005, Seite 5

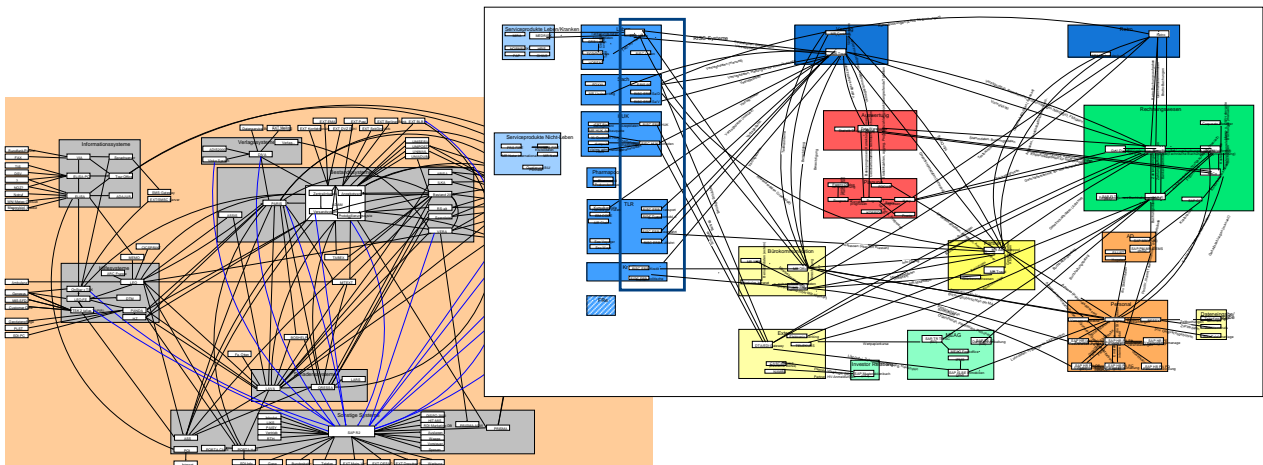
Die technische Architektur der SOA-Infrastruktur ist von der fachlichen Architektur der IT-Landschaft zu trennen

- Die Services der SOA sind fachliche *Enterprise Services*
- *Technische Services* der SOA-Infrastruktur gewährleisten ihre Benutzbarkeit
- *Technische Services* innerhalb der Infrastruktur bestimmen die Eigenschaften der *Enterprise Services*
 - Interaktionsstile: request/reply, one-way-message, publish/subscribe
 - Security: Authentifizierung, Verschlüsselung, ...
 - Safety: Fehlersemantik, „Transaktionssicherheit“
 - Binding: Adressierung des Service, Auffinden des Service Providers
 - ...

- Die Entwicklung einer SOA-Infrastruktur ist klassisches Software Engineering

sd&m AG, 22.07.2005, Seite 6

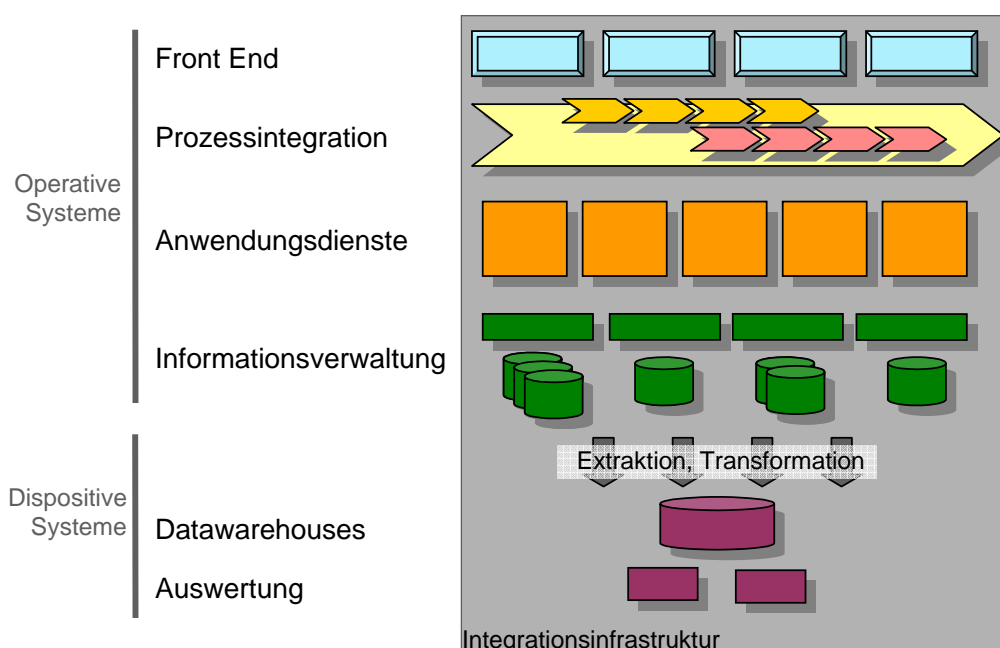
Voraussetzung für die Wiederverwendung fachlicher Services ist eine fachliche Architektur der IT-Landschaft



- Lose Kopplung verlangt nach Trennung der Zuständigkeiten
- Strukturierung und Trennung von Innen- und Außensicht notwendig
- Diese SE-Prinzipien sind auf die technischen und fachlichen Strukturen anzuwenden

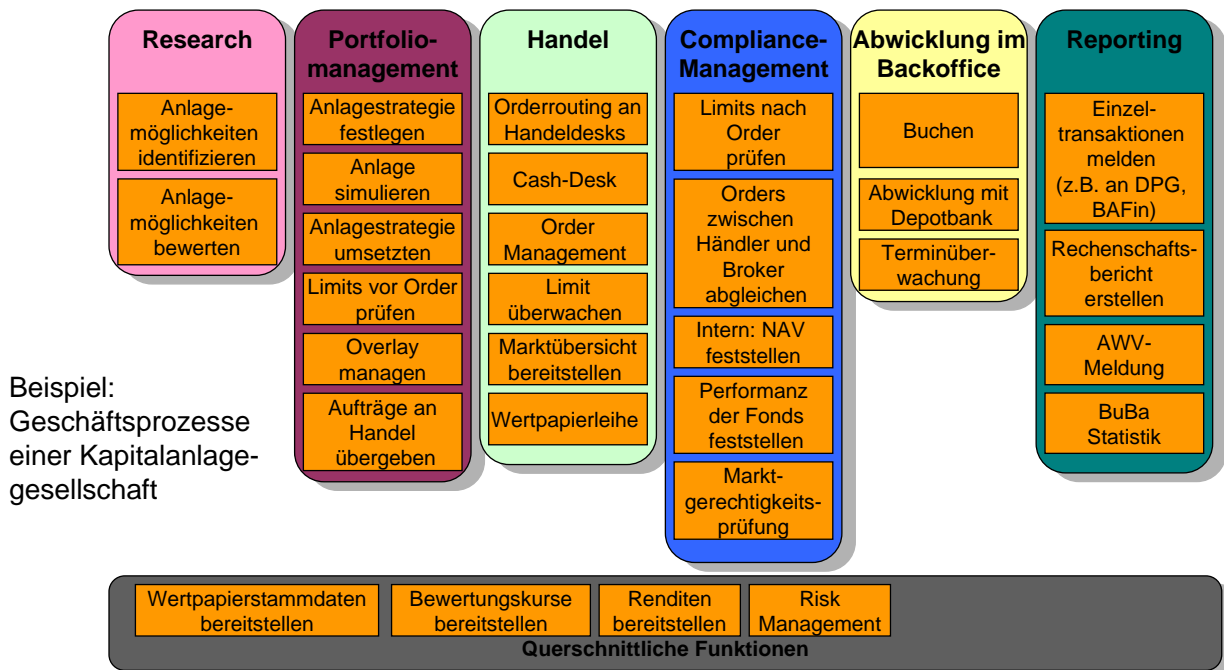
sd&m AG, 22.07.2005, Seite 7

Eine vertikale Strukturierung erfolgt anhand der primären Aufgabe in der DV-Verarbeitungskette



sd&m AG, 22.07.2005, Seite 8

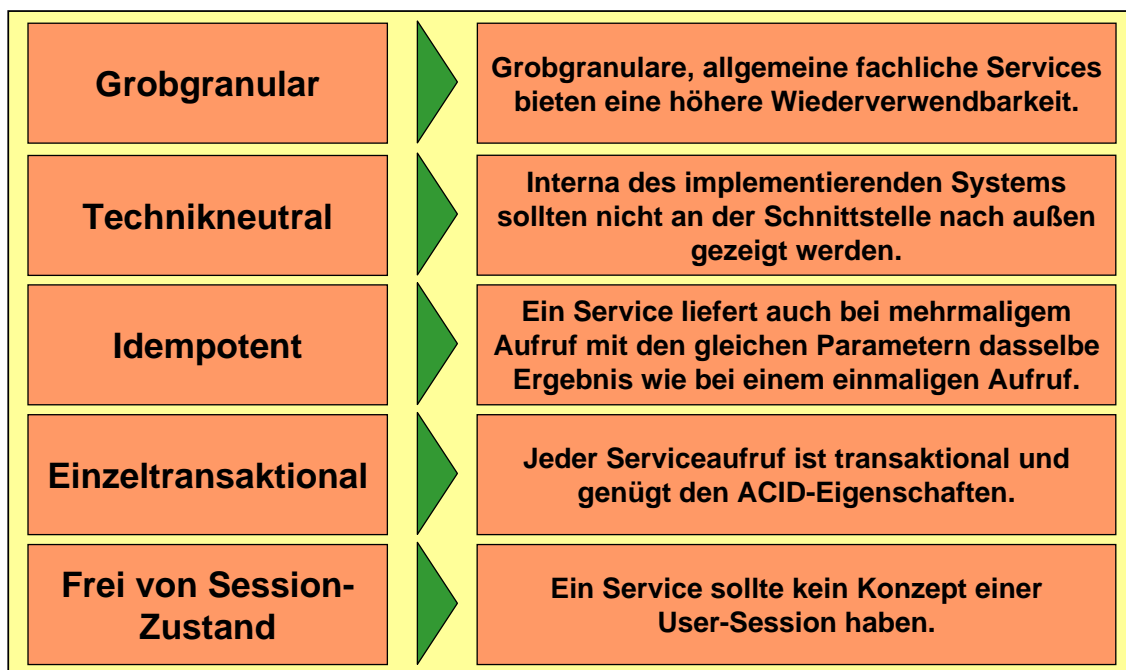
Die Bildung fachlicher Komponenten basiert auf der Analyse der Geschäftsprozesse



sd&m AG, 22.07.2005, Seite 9

Service-Schnitt

Service-Eigenschaften tragen dazu bei, Anwendungssysteme lose zu koppeln



sd&m AG, 22.07.2005, Seite 10

SOA verlangt enge fachliche Zusammenarbeit zwischen IT-Organisation und Fachabteilungen

- Service Governance ist fachliche und technische Aufgabe
 - Serviceportfolio muss an Anforderungen des Unternehmens ausgerichtet sein
 - Umfassende und einheitliche Dokumentation unabdingbar
 - Fortentwicklung der SOA muss übergreifend geplant werden
 - Sich entwickelnde Standards können zur Kostensenkung beitragen
- Die IT-Organisation braucht zentrale und dezentrale Gliederungen
 - Architektur-Board zur Vereinheitlichung und Abstimmung
 - Fach-IT-Abteilungen mit hohem fachlichen Know-How bei der Definition und Realisierung von Services

sd&m AG, 22.07.2005, Seite 11

Fazit

- Die Migration zu einer Serviceorientierten Architektur betrifft die gesamte IT
 - SOA ist kein Technik-Projekt
 - SOA setzt aktive Gestaltung der Architektur der Anwendungslandschaft voraus
 - SOA baut auf enge Zusammenarbeit zwischen IT- und Fachabteilungen
- SOA ist eine strategische Entscheidung des Unternehmens
 - Unterstützung aus der Geschäftsleitung notwendig

sd&m AG, 22.07.2005, Seite 12



**Vielen Dank für Ihre
Aufmerksamkeit**

sd&m AG, 22.07.2005, Seite 13

Wozu SOA?

Dr. Markus Pizka
Technische Universität München
Institut für Informatik
Prof. Dr. Dr. h.c. Manfred Broy

Motivation

- bis 99: Dr. rer. nat, Verteilte Betriebssysteme
- seit 03: Spin-Off itestra GmbH
- seit 01: Software Engineering
 - Kompetenzzentrum Software Maintenance
 - 80% Kosten ...
 - Workshop Reengineering 12/04 ~ *SOA ist Lösung!*

Service Oriented Architecture

-- Flexibilität und Wiederverwendung



Simplify and support change with a service-oriented architecture.

Begründung?

Thesen



SOA

- ist nichts wirklich Neues,
- ändert die Wartungsprobleme nicht und
- schafft neue Probleme...



Alternativen:

- Qualitätsmanagement.

21.07.2004

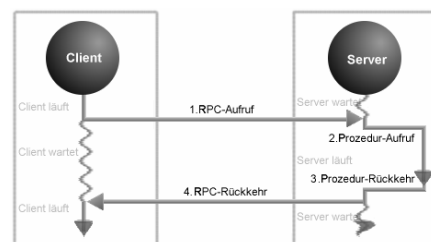
Dr. Markus Pizka

3

Was ist ein Dienst?



- SOA ~
Architekturkonzept, Menge von Services, Bereitstellung fachlicher Funktionalitäten in Form von Services
- Services ~
Funktionalität, def./Standard Schnittstelle, lose gekoppelt, Nachrichten, wieder verwendbar, unabhängig
- Wünsche, keine Tatsachen
- Nicht neu:
 - Math, Funktionen: $f(x) = x^2$??
 - OO Klasse / Methode
 - SUN Remote Procedure Call (80er)
- Unrealistisch
 - **unabhängig** \Rightarrow kein Zusammenwirken in einem System?



21.07.2004

Dr. Markus Pizka

4



- **System:**
 - aus Komponenten zusammengesetzte Einheit
- es gibt „innen“ und „außen“ (Umwelt)
- **offen:** Wechselwirkungen mit Umwelt
- **dynamisch:** Eigenschaften können sich ändern
- **Komponente:**
 - aus Komponenten zusammengesetzte Einheit mit „innen“/„außen“
 - atomar (unteilbar, vordefiniert)
- **Eigenschaften der Komponenten**
 - innere Operationen
 - äußere Operationen (**Schnittstelle** zur Umwelt)
- **Zusammenwirken** ⇔ **Abhängigkeiten** ⇔ **Struktur**

21.07.2004

Dr. Markus Pizka

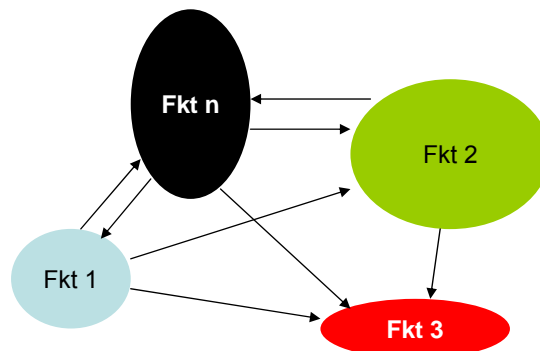
5

Abhängigkeiten - Vorstellung



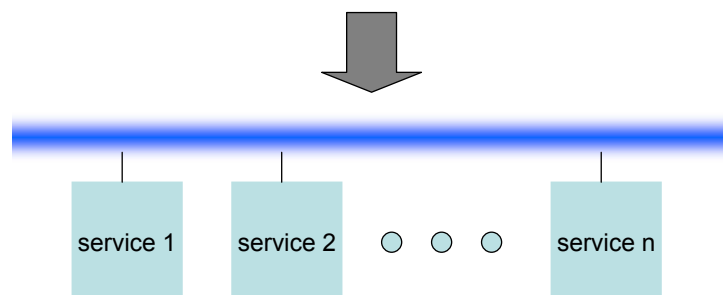
Alt (Legacy)

- heterogen
- vielfältige, intensive Abhängigkeiten
- Monolith!



Neu (SOA)

- homogen
- wenige Abhängigkeiten



21.07.2004

Dr. Markus Pizka

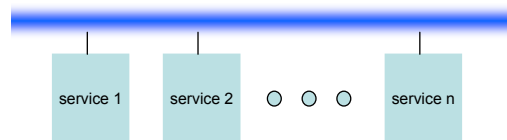
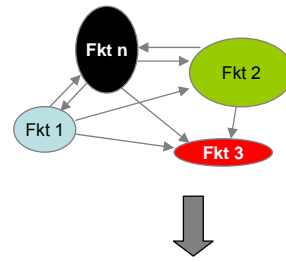
6

Bedeutung der Kanten

Nutzungsabhängigkeiten
(Aufrufbeziehungen, Daten)

≠

Kommunikationsmedium



Relevante Abhängigkeiten

- explizit: Daten-/Kontrollfluss, Vererbung, Schachtelung, ...
- implizit: Semantik, Daten, Annahmen
- ~ *Separation of Concerns* (multidimensional)

21.07.2004

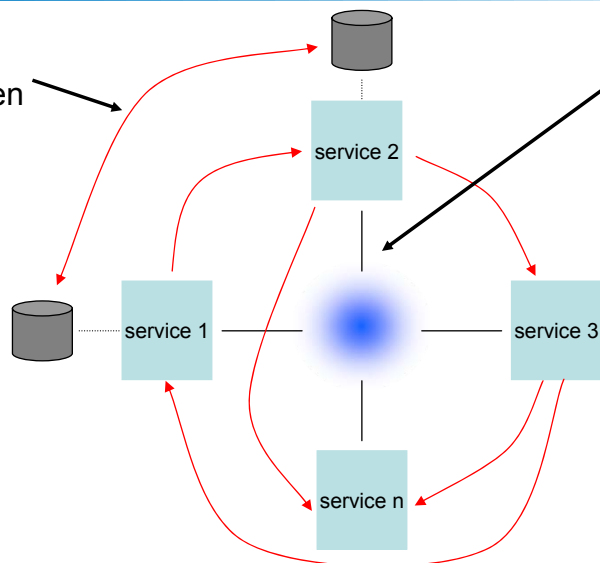
Dr. Markus Pizka

7

Realität

Inhaltliche
Abhängigkeiten

Nachrichten-
kommunikation



- Abhängigkeiten reduzieren Flexibilität!
- ~~Architektur~~ ⇒ Struktur der Abhängigkeiten

21.07.2004

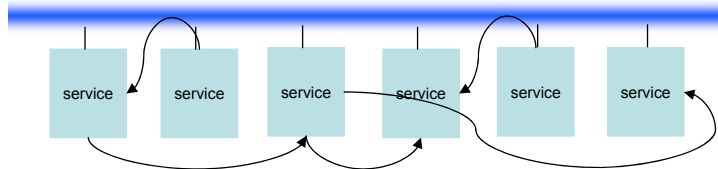
Dr. Markus Pizka

8

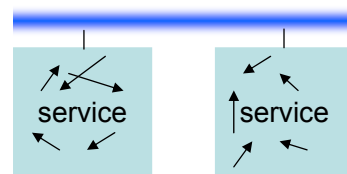
Granularität – Umfang der Services



- Feingranular
 - zahlreiche Einzelservices
 - hohe Flexibilität / Wiederverwendbarkeit
 - **intensive äußere Abhängigkeiten**



- Grobgranular
 - wenige, umfangreiche Dienste
 - **sinkende Stabilität / Kompositionalität**
 - **intensive innere Abhängigkeiten**



- Strukturproblem inhärent!

21.07.2004

Dr. Markus Pizka

9

Wartbarkeit - Kernprobleme



- C.S. Hartzmann 93. „Maintenance Productivity“
 - T1 – Problem-Verständnis
 - T2 – administrative Zeit
 - T3 – Werkzeugbereitstellung
 - T4 – Analyse
 - T5 – Spezifikation der Änderung
 - T6 – Kodierung
 - T7 – Unit Test
 - T8 – Integrationstest
 - T9 – Review
- Faktoren: vielfältig
 - Organisation: Skills, Team, Kommunikation, Management, Prozess, ...
 - Techniken: Doku, Style Guides, Modellierung, Generatoren, Sprache, ...

ca. 50% Verstehen!

21.07.2004

Dr. Markus Pizka

10

Beispiel Code-Cloning

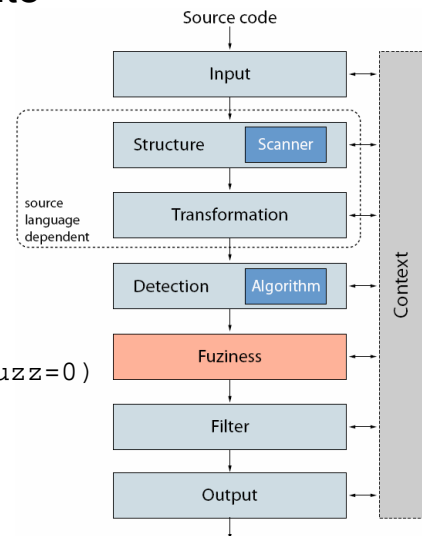


■ Äquivalente Programmfragmente

- Copy+Paste, Frameworks
- Änderung an multiplen Stellen
- Anzahl, Ort unbekannt

■ TUM Clone-Detective

- Informationssystem 800kloc < 20s
- 2816 Klone ≥ 10 StmtS
- #1 Clone class with 2 instances(fuzz=0)
ComplDep.cc (640) len:246
ComplPure.cc (624) len:245
- Vermeidung statt Beseitigung



21.07.2004

Dr. Markus Pizka

11

Neue Probleme



1. Performance

- Distributed Memory, Multiple Instruction Multiple Data (MIMD)
- Nachrichtenaustausch (ca. 10^5 langsamer als Speicher)
- Probleme s. HPC Forschung 90er Jahre
- „Join“ über verteilte Datenhaltung?

2. Orientierung an Geschäftsprozessen

- Architektur anhand GPs: instabile (teure) Systeme
- Nichts ist **volatiler** als GPs!
- 55% CRs: veränderte GPs
- Orchestration \Rightarrow Verlagerung
Services völlig unabh. von Änderung?



McIDAS (seit 1973):

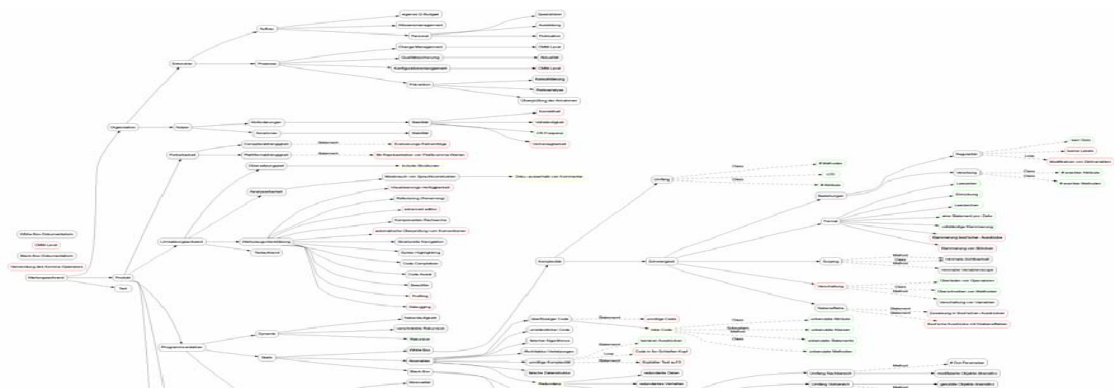
„Requirements are not our final goal, user satisfaction is.“

21.07.2004

Dr. Markus Pizka

12

Alternative: Qualitätsmanagement



- TUM Qualitätsmodell „Wartungsaufwand“
- ca. 250 Kriterien, (semi-)automatische / manuelle Prüfung
- zeitnahe Kontrolle
- Vermeidung statt Beseitigung
- Bsp.: überflüssig, Homonyme, reguläre Strukturen, Fluktuation, ...

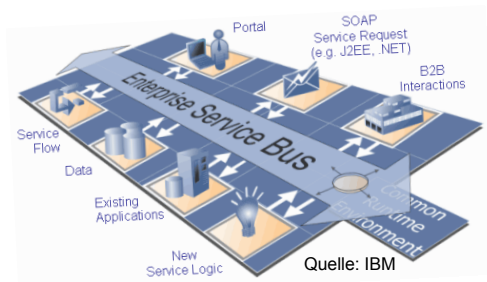
21.07.2004

Dr. Markus Pizka

13

Zusammenfassung

- Kritik
 - Erhöhte Wartbarkeit, niedrige Kosten nicht nachvollziehbar
 - Kernprobleme werden nicht adressiert
- Positiv
 - Denken in separierbaren Einheiten
Fördert u.U. *Separation of Concerns*
 - Technische Entkopplung, z.B. separate Produktion
- Marketing
 - Breite Produktpalette
 - Konzeptioneller Fortschritt?



21.07.2004

Dr. Markus Pizka

14

Modellbasierte dienstorientierte Software- Entwicklung

Johannes Grünbauer

gruenbau@in.tum.de



Technische Universität München
Institut für Informatik

Lehrstuhl Prof. Dr. Dr. h.c. Manfred Broy



Überblick

- Motivation
- Projekt MEwaDis
- Dienste / Dienstbegriff
- Abgrenzung zu Komponenten
- Dienstbasierter Entwicklungsprozess
- Adaptivität
- Kontext
- Fallstudien

Motivation

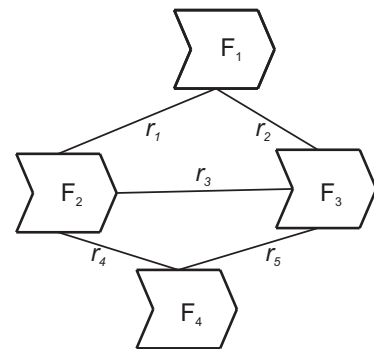
- Gesamtkosten für Elektronik in Fahrzeugen der Oberklasse betragen bis zu 40% der Gesamtkosten
 - Funktionen sind über verschiedene Steuergeräte verteilt und mit verschiedenen Bus-Systemen verbunden
 - Problem der „unwanted feature interactions“
 - Unterschied zu klassischen Anwendungen: Interaktionen und Abhängigkeiten
 - Multifunktionales System
 - ♦ Vielfache Interaktion von Funktionen
 - ♦ Starke wechselseitige Beziehungen
 - ♦ Funktionsabhängigkeiten
- ⇒ Dienstbasierte Modellierung

Projekt MEwaDis

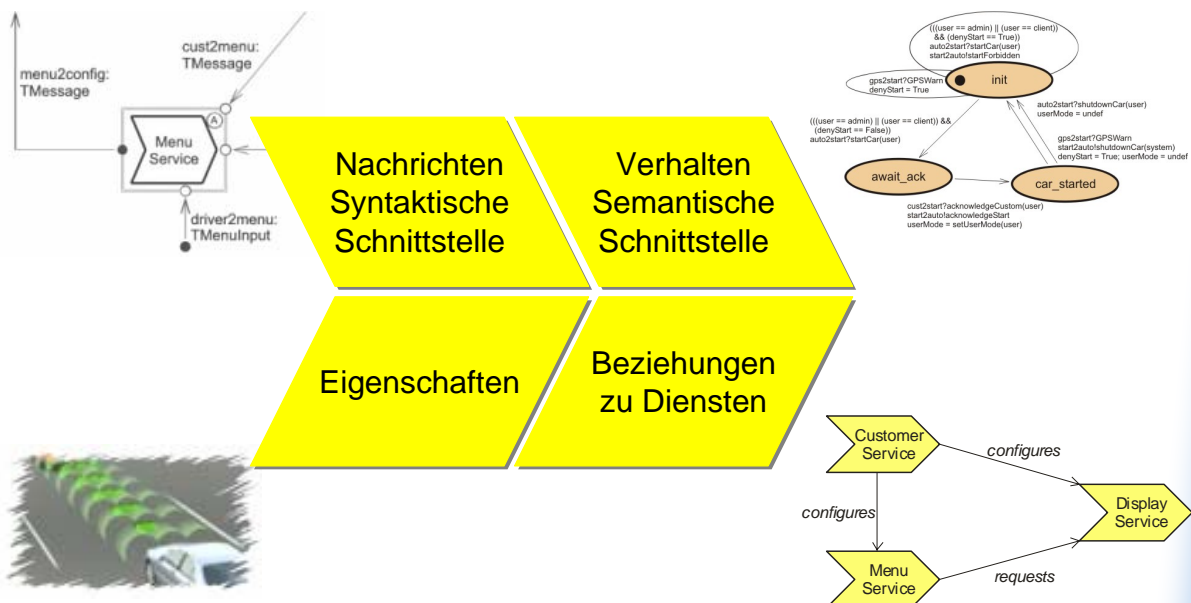
- Modellbasierte Entwicklung adaptiver Dienste
 - ♦ Systemmodell: Struktur, Verhalten + Kontext
 - ♦ Aussagenmodell: System ↔ Kontext□
 - ♦ Formalisierter Dienstbegriff
 - ♦ Verifizierbares Sicherheitsmerkmale (Security, Safety)
 - ♦ Methodischer Entwicklungsprozess
- Anwendung
 - ♦ Man-Machine-Interface, adaptive Dienste in Fahrzeugen
 - ♦ Prototypische Implementierungen
- Kooperationspartner
 - ♦ BMW F&T, BMW Car IT GmbH (im Rahmen der High-Tech Offensive Bayern)

Dienste

- Formalisierung durch Interaktionsmuster
- Partielle Definition (im Gegensatz zu einer Komponente)
- Dienste als Modellierungseinheit
- Eine Komponente kann mehrere Dienste enthalten
- Ein Dienst kann in unterschiedlichen Kontexten agieren
- Die Kombination von Diensten ergibt eine „logische Dienstarchitektur“



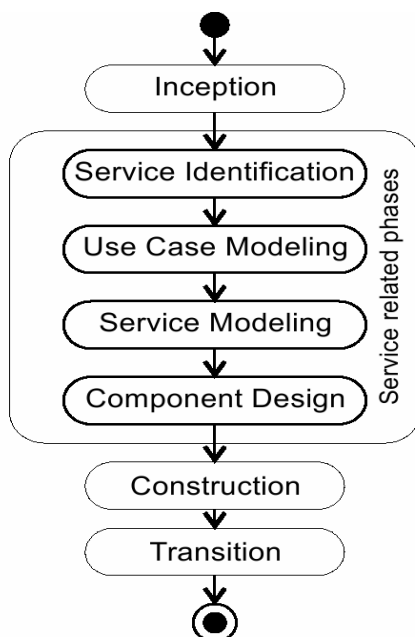
Dienstbegriff



Abgrenzung zu Komponenten

- Konzeptuelle Unterschiede
 - ◆ Höhere Abstraktionsebene, „logische Dienstarchitektur“
 - ◆ Keine Dekomposition
 - ◆ Partielle Verhaltensspezifikationen
- Fokus auf Interaktionen
- Konkrete Service-Architekturen:
 - ◆ Web-Services, JINI, CORBA, OSGi

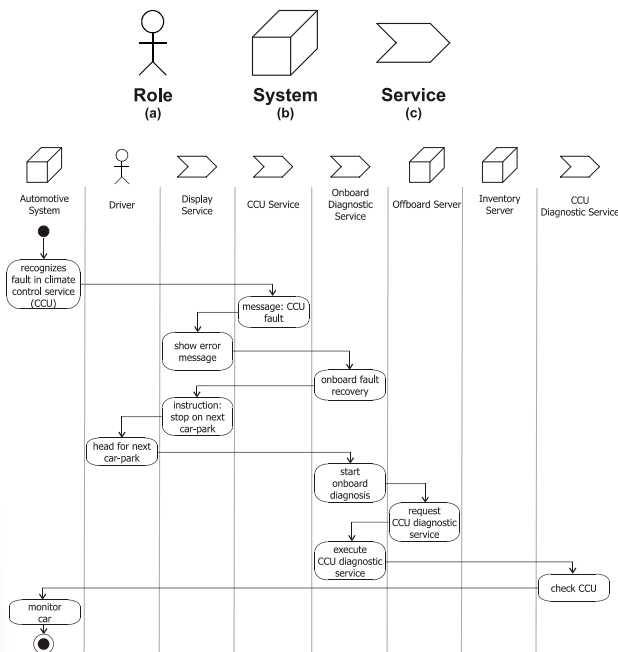
Entwicklungsprozess



- Modellbasierter, phasenorientierter Prozess
- Basiert auf Unified Process
- Inkrementell
- Dienst-spezifische Teilprozesse in der *Elaboration Phase*

- Dienste als First-Level Entwicklungsobjekte

Dienst Identifikation



- Grundlagen:
 - ♦ Projektziele
 - ♦ Produktfunktionen (z.B. aus der Anforderungsspezifikation)
- Erweitertes Aktor Modell
 - ♦ Rollen und System als Aktoren
 - ♦ NEU: Dienste als Aktoren
- Anforderungen in Aktivitätsfluss wandeln
 - ♦ Sequenz von Aktivitäten
 - ♦ Aktivitäten den Aktoren zuordnen (Swimlanes)

Use Case Modellierung

- Ausarbeiten der Use Case Beschreibungen
- Spezifikation von Bedrohungen und Sicherheitsaspekten
- Identifikation der Dienst-Beziehungen
- Formalisierung der Use Case Beschreibungen in Sequenzdiagrammen

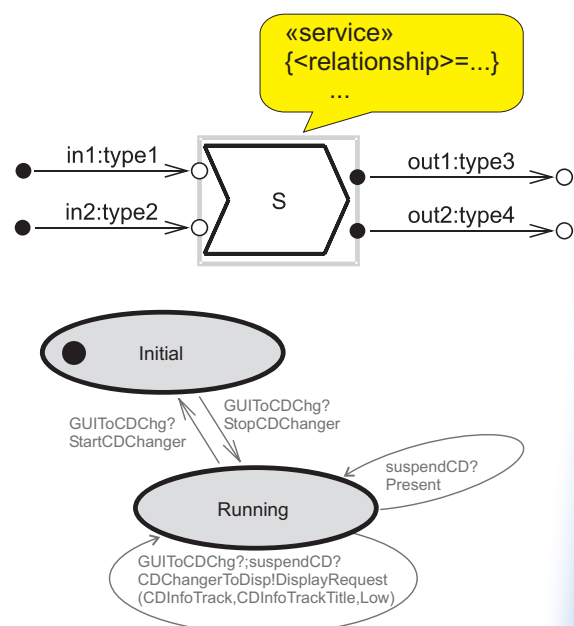
Use Case Modellierung

Feld	Beschreibung
Use Case	Name des Use Cases
Aktoren	Welche Aktoren sind betroffen?
Vorbedingung	für den Use Case
Ablauf	Was passiert im Use Case?
Varianten	Was kann noch passieren?
Eingabe	Welche Eingaben werden verarbeitet
Ausgabe	Welche Daten werden ausgegeben?
Sicherheit	Welche Sicherheitsziele
Beteiligte Dienste	Welche anderen Dienste sind betroffen?

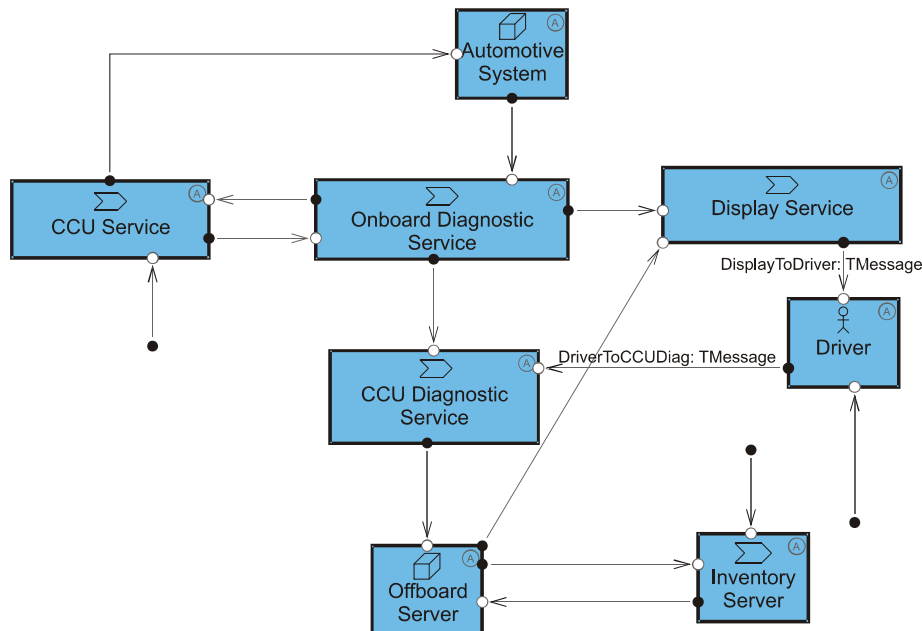
Modellierung / Tool Support

AutoFocus

- Grafische Spezifikation von verteilten Systemen
- Basiert auf der formalen Methode *Focus*
- Verschiedene Sichten
 - ♦ Systemstrukturdiagramme
 - ♦ Zustandsübergangsdiagramme
 - ♦ Datentyp-Definitionen
 - ♦ Extended Event Traces
- Konsistenzüberprüfungen
- Verifikation des Modells



Modellierung / Tool Support

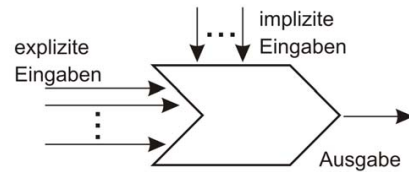


Modellierung / Tool Support

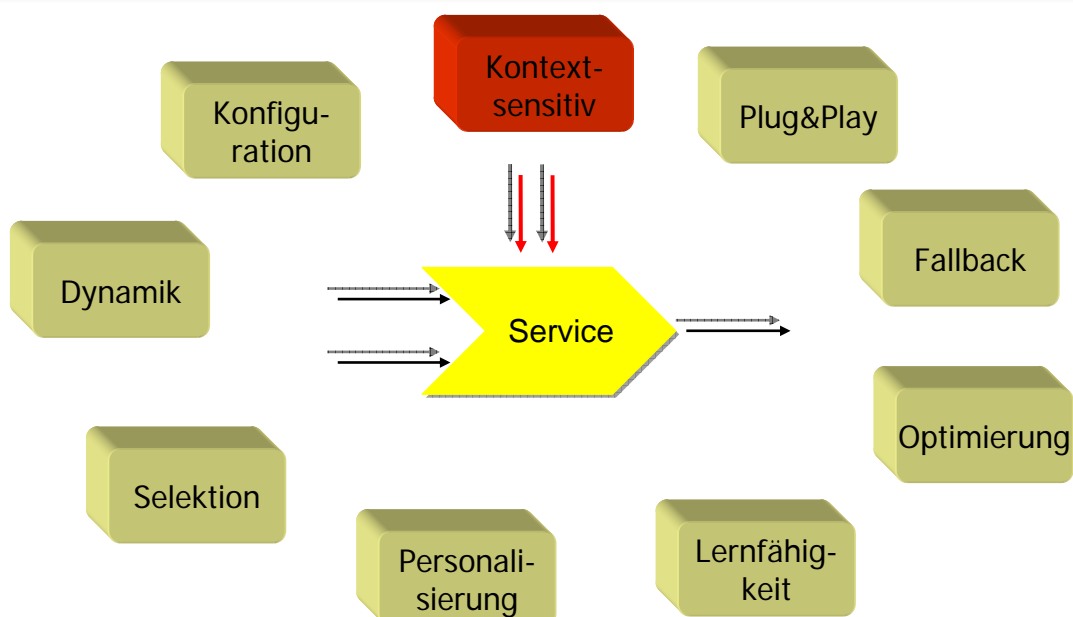
- Modellverifikation mit SMV
 - ◆ Verifizierung von Eigenschaften von Safety und Security
 - ◆ AutoFocus erzeugt Eingabe für SMV
 - ◆ Verletzung der Eigenschaften visualisiert
- Codegenerierung aus AutoFocus
 - ◆ Java, C, Ada, etc.
 - ◆ Generierung lauffähiger Dienste für OSGi-Framework

Adaptivität

- Dynamische Anpassung von Diensten an die Umgebung (Kontext)
- Erweiterung der Eingabekanäle
 - ◆ Explizite Eingabekanäle
 - ◆ Implizite Eingabekanäle
- Kontextneutralität

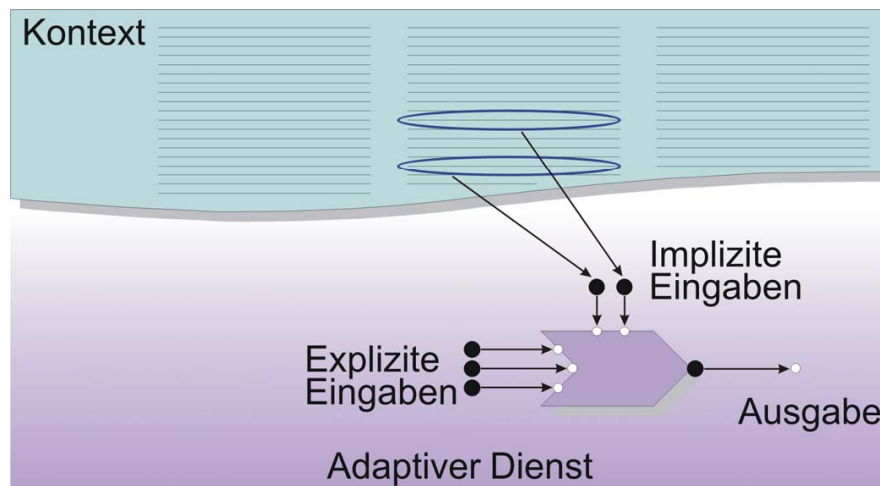


Eigenschaften adaptiver Dienste



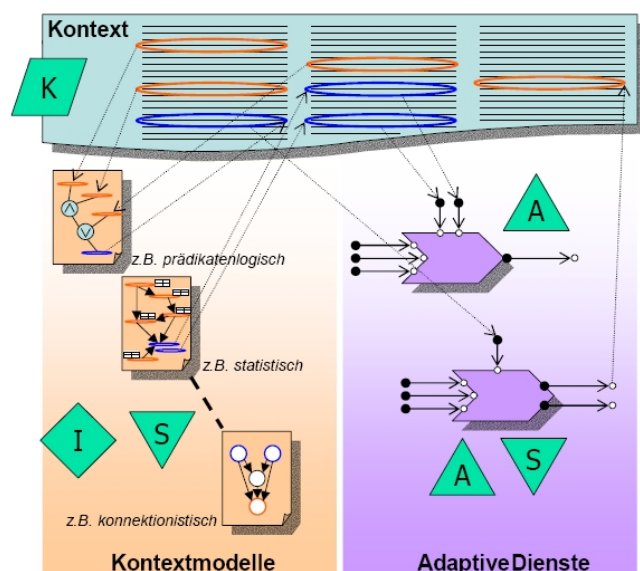
Kontext

Kontext ist jegliche Information, die dazu benutzt wird, den Zustand von Entitäten zu charakterisieren, die relevant für Interaktionen sind.



Adaptiver, kontextsensitiver Dienst

- Unterscheidung der Eingaben
- Trennung zwischen Kontext und Kontextmodell
- Dienstrollen (Sensor, Kontext, Interpret, Aktuator)
- Unterscheidung von Kontextmodellen



Fallstudien

- Automotive Systems Lab
- Klimaanlage-Diagnose
- Adaptiver Anzeigendienst
- Adaptive Benutzerführung
- Authentifikation des Fahrers im Fahrzeug

Zusammenfassung

- Hohe Komplexität von Systemen
- Requirements Engineering
- Korrekte Modellierung von verteilten Systemen
- Erkennung von Feature Interactions

- Veröffentlichungen:
<http://www4.in.tum.de/~mewadis>



SOA – aber wie? Nutzen, Management, Bauanleitung?

Inge Hanschke
21. Juli 2005



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 1

Übersicht

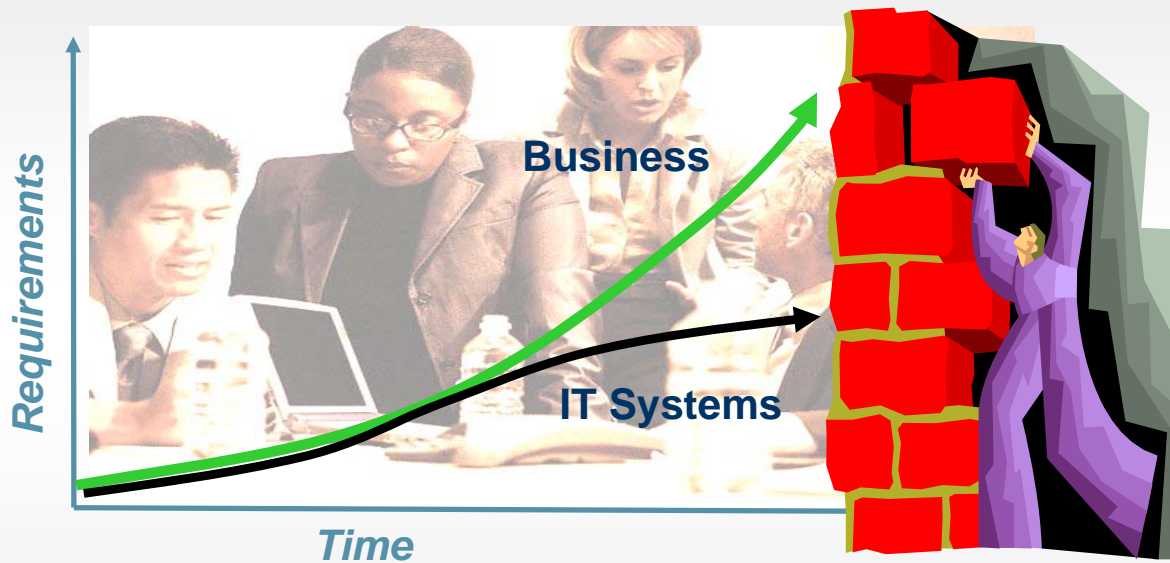
- Motivation
- Was ist SOA?
- SOA Governance
- Fazit



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 2

The IT Reality



Constant changes reduces the ability to respond

Ken Wilner : „SOA Elucidated: Principles of Service-oriented Architecture “

Flexibilität durch Service-orientierte Architekturen?

- Business-Anforderung:
 - Flexibilität in der Service-Bereitstellung
 - Fähigkeit schnell auf verändernde Marktanforderungen zu reagieren
 - Kostengünstige Bereitstellung
- SOA als eine Antwort zum Business Alignment?
 - Flexibilität – „Ability to change“
 - Wiederverwendung
 - Erweiterbarkeit / Offenheit

Business Opportunity

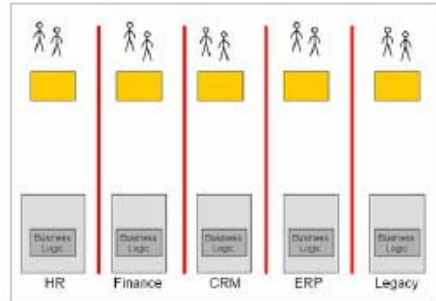


Figure 1 - Business today is limited by software silos



Figure 2 - SOA represents a business opportunity

By Motti Vaknin : „The Business Imperative of SOA Governance “



Service oriented architecture

- “SOA is an architectural style for building software applications that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services. A service is an implementation of a well-defined business functionality, and such services can then be consumed by clients in different applications or business processes.”

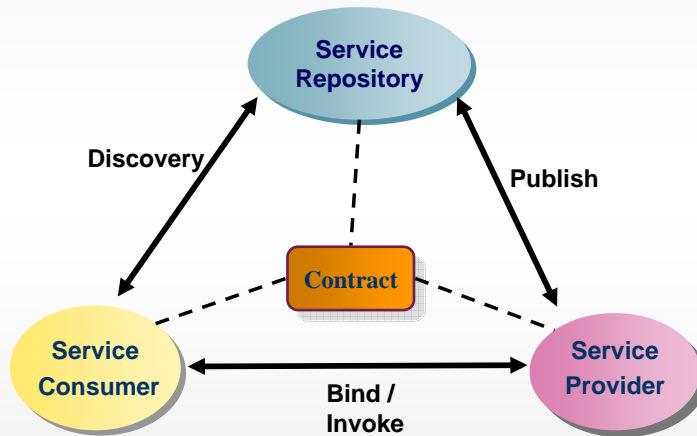
Qusay H. Mahmoud, Sun Microsystems, April 2005

Services	➔	Softwarekomponenten mit Kontrakt
Grobe Granularität	➔	Fachliche Einheiten
Lose gekoppelt	➔	Unabhängigkeit der Services
Implementationsunabhängig	➔	Trennung von Schnittstelle und Implementierung
Dynamische Discovery	➔	Services sind selbstbeschreibend
SOA Infrastruktur	➔	Integrationsinfrastruktur mit Services



SOA Modell

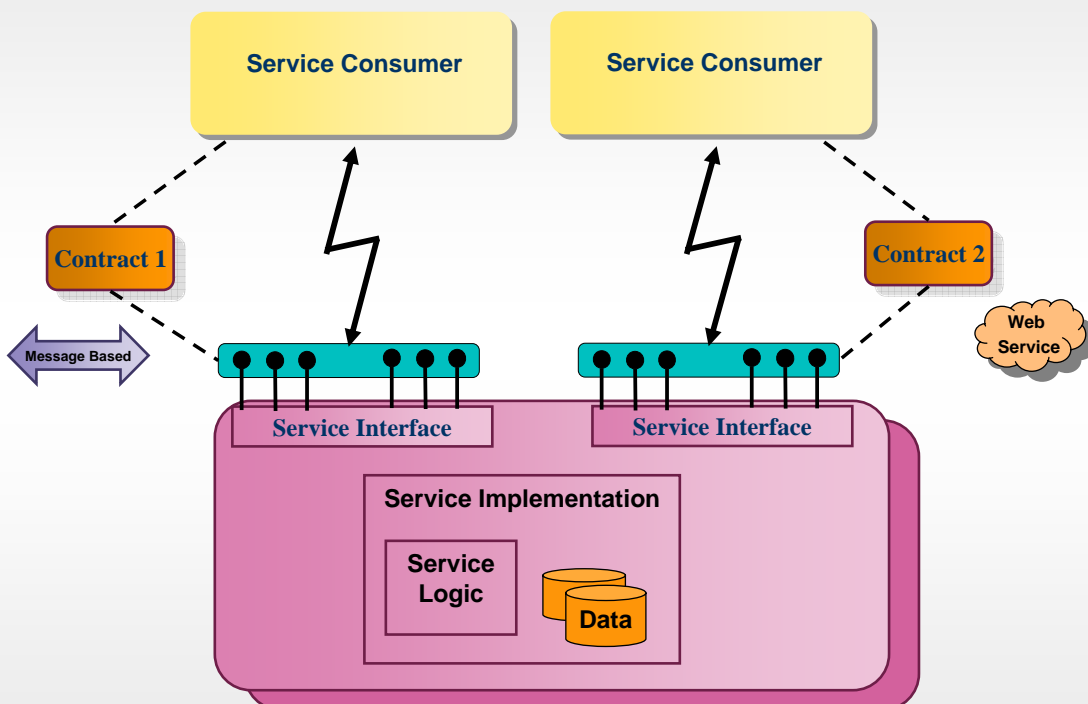
- Lose gekoppelte Komponenten, die über wohldefinierte Schnittstellen kommunizieren



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
 © iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 7

SOA Modell



Ken Wilner : „SOA Elucidated: Principles of Service-oriented Architecture “

Service Klassifikation

	Basic Service	Intermediary Service	Process-centric Service
Charakteristika	Einfach datenzentrierte oder logikzentrierte Services; Ausschließlich Service Provider	Aggregieren Basic Services; Sowohl Service Provider als auch Service Consumer	Prozesslogik in Service gekapselt häufig mit BPMS umgesetzt – steuern Kernbusinesslogik-Services; Sowohl Service Provider als auch Service Consumer
Anwendungsfälle	Fachlicher (Domain-spezifisch) oder technischer Service (Security, Logging, Persistence, Auditing)	Konzeptionelle oder technische „Bridge“; Technologische Gateways, Adapter, Fassaden; „functionality-added“ Services	Applikationsspezifische Ablauflogik / Prozesse
Eigenschaften	Stateless, ACID-Transaktionen, QoS z.B. Security auf Basis der einzelnen Services	Stateless, ACID- und tw. Langlaufende Transaktionen (Kompensationstransaktionen), QoS für die aggregierten Basis Services	Stateful - Übergreifendes Zustandsmanagement; Langlaufende Transaktionen (Kompensationstransaktionen); QoS für die eingebundenen Services

SOA Governance

- Sicherstellung, dass die Business Ziele / Anforderungen auf strategischer und operativer Ebene umgesetzt werden
- IT als Enabler – Schaffung neuer Möglichkeiten und Nutzen
- Verantwortlicher Umgang mit den IT Ressourcen
- Angemessenes Management von IT-Risiken

- Organisation, Regeln, Prozesse und Metriken zur Unterstützung von Planungs-, Erstellungs-, Entscheidungs- und Controllingprozessen

- „What to do.“
- „How to do.“
- „Who should do it.“
- „How it should be measured.“



SOA Governance - Kernfragen

- Entscheidung für SOA?
 - Analyse der Unternehmensstrategie und der Ausgangslage
 - Welche Prinzipien und Leitlinien sind erforderlich, um das Business Alignment der IT zu fördern?
 - Was sind angemessene Strukturen für das Zusammenspiel des Business mit der IT? Wie erreicht man Konsistenz und eine ausreichende Flexibilität für den Umgang mit Business Änderungen?
 - ...



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 11

Business Alignment der IT

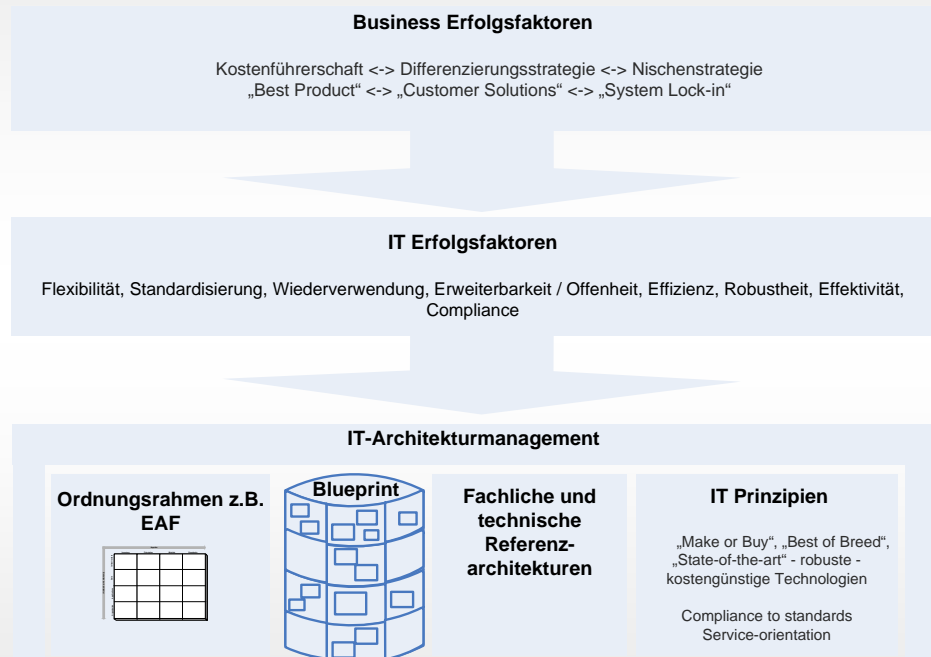
- Businessstrategie – „Differenzierungspotential verstehen“
- Ableiten der Business Erfolgsfaktoren z.B. Kostenführerschaft
- Ableiten der IT Erfolgsfaktoren z.B. Flexibilität
- Umsetzung durch IT-Architekturmanagement
 - Standardisierung z.B. Blueprint
 - Ordnungsrahmen z.B. Enterprise Architecture Framework
 - fachliche und technische Referenzarchitekturen
 - strategische IT-Prinzipien z.B. „Best-of-Breed“



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 12

Business Alignment der IT



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
 © iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 13

SOA Lösungsansätze

Problemstellungen	Mögliche SOA Lösungsansätze
Business Agilität	Best Practices, offene Standards, lose gekoppelt und gemeinsam genutzte Services
„time to market“	Wieder verwendbare Business Services, Serviceorientierte Prozesse (kleiner Anteil von Neuentwicklungen, kurze Entwicklungszyklen)
„End-to-End“ Sicherheit	Identity Management, Security Standards (z.B. WebServices)
Vermeidung von „vendor lock-in“	Offene Industrie Standards, die die Interoperabilität und Plattformunabhängigkeit ermöglichen
Kostenreduktion	inkrementelle Projekte (schneller ROI), Wiederverwendung von Prozessen und SOA-Infrastruktur, agile Prozesse



Jay Shah: „Service-Oriented Architecture (SOA) Primer II“; Gartner 2005

SOA – aber wie? – Inge Hanschke – 21. Juli 2005
 © iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 14

SOA Governance - Kernfragen

- SOA – aber wie?
 - Wie wird eine SOA organisatorisch eingeführt und umgesetzt?
 - Wie sieht das Management der Services aus?
 - Welche Standardisierung für Services, Service Beschreibungen und Zusammenspiel der Services ist angemessen?
 - Wie kann das Controlling von Services und Service Providers durchgeführt werden? Wer sollte das Monitoring und Änderungsentscheidungen für die vorhandenen Services durchführen?
 - Welche Probleme bestehen in der Nutzung von Services und wie sollte der Support dies unterstützen?
 - ...



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 15

SOA Governance

- Neue Paradigmen
 - Domain Ownership
 - Develop now, Integrate later” -> “Develop for Integration”
- Grundsätze und Leitlinien
 - Business Level z.B. Adressinformationen dürfen nicht als Klartext übermittelt werden
 - Sicherheit z.B. Nachrichten müssen ein „Authorization Token“ beinhalten
 - Softwarearchitektur z.B. Service Design
- Sicherstellung der Einhaltung der Grundsätze
- Qualitätssicherung und kontinuierliche Optimierung
 - Werkzeug-Unterstützung z.B. zur Überprüfung der Einhaltung der Vorgaben
 - Prozesse



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 16

SOA Governance

- Prozesse
 - Servicedefinition eingegliedert in das Bebauungsmanagements (landscape modelling)
 - Managementprozesse als Bestandteil der IT-Projektportfolioprozesse insb. Multiprojektmanagement
 - Servicebereitstellung
 - Serviceentwicklung
 - Servicewartung & -support
- Organisation eingebettet im IT-Architekturmanagement



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 17

SOA Rollen

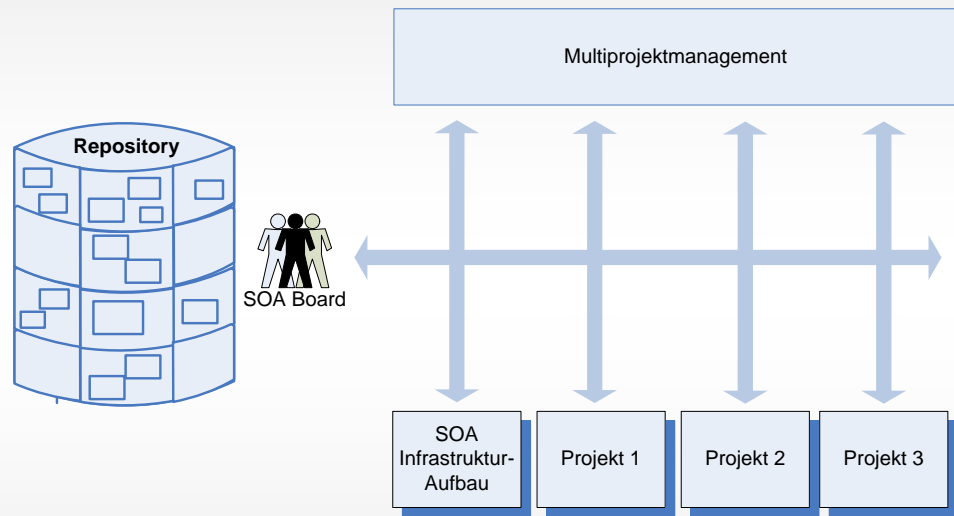
- SOA Board
 - Inhaltliche Festlegung von Services (Kontrakte, Repository)
 - Backbone für das Multiprojektmanagement
- SOA Architekturteam
 - Architekturfestlegungen, Prinzipien, Leitlinien und Standards
 - Inhaltliche Festlegung zu Werkzeugen und Service Infrastruktur
 - Aufsetzen des Servicedefinitionsprozesses
 - Coaching, QS und Reviews
- SOA Infrastrukturteam
- Domain-orientierte Rollen
 - Domain Owner
 - Service oriented business analyst & Line of business representative
 - Developer & Maintainer
 - Service Tester



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 18

SOA Board



SOA – Verschiedene Umsetzungsalternativen Bottom-up <-> Top-down

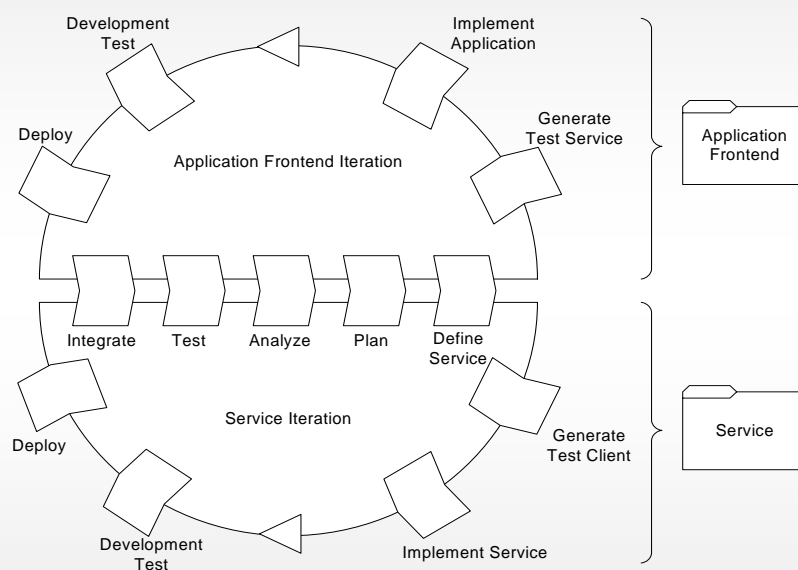
	Bottom-up	Top-down
Charakteristika	„Projektgetrieben“ Entwicklung eines Service wird durch Projektanforderungen initiiert	Unternehmensübergreifend Services werden im Rahmen vom Bauungsmanagement / IT-Architekturmanagement definiert
Vorteile	Verwendbarkeit sichergestellt Erfahrungen werden „im Kleinen“ gesammelt	Business-Alignment Design für unternehmensübergreifenden Einsatz Risiko redundanter Services reduziert Nutzen von Investitionen leichter transparent zu machen
Herausforderungen	Berücksichtigung von strategischen und unternehmensübergreifenden Aspekten Projekt-Granularität für Enterprise Level passend? Siloed SOA führt nicht zwangsläufig zu einer Enterprise Architektur	Komplexität (Anforderungen, Servicedefinition, -entwicklung und -bereitstellung) Unternehmensübergreifendes SOA erfordert eine hohe Anfangsinvestition

Serviceorientierung und Projektmanagement

- Erster Service-Design muss Teil der Projektdefinition sein:
 - ➔ Application Frontends, externe Services, Basis Services
 - ◆ Welche Services müssen neu entwickelt werden? Welche basieren auf bestehenden Anwendungen?
 - ◆ Welche Erweiterungen oder Modifikationen müssen für bestehende Services durchgeführt werden?
- Intermediary und Process-centric Services können später designt werden
- Service Kontrakte als Heartbeat für die Projekte:
 - ➔ Projektkosten und Zeitabschätzungen
 - ➔ Inkremente
 - ➔ Projektsynchronisation
 - ➔ Projekttest und Rollout-Planung



Service Contracts als Heart Beat von Projekten



Dirk Krafczig, Karl Banke, Dirk Slama: „Enterprise SOA: Service Oriented Architecture Best Practices“



SOA Prinzipien und Leitlinien

- Prinzipien

- Services sind das Hauptstrukturierungselement
- Für jeden Service wird ein formaler Kontrakt definiert
- Serviceinteraktion erfolgt nur über wohldefinierte Schnittstellen
- Services sind nur über die SOA-Infrastruktur ansprechbar
- Services müssen so geschnitten sein, dass sie eine fachliche Bedeutung für den Anwender haben
- Services müssen grob granular sein
- Services müssen lose gekoppelt sein



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 23

SOA Prinzipien und Leitlinien

- Leitlinien

- Prozessintegrität (Technical Failure, Business Exceptions)
- Technische Lösungen
 - ◆ Logging und Tracing
 - ◆ ACID Transaktionen
 - ◆ Fachliche Transaktionen (nested, multilevel, Sagas, Kompensation, ...)
- Empfehlungen für SOA Architekten
 - ◆ Optimistisches Sperren als Default
 - ◆ Update Operations idempotent
 - ◆ Vermeide verteiltes 2PC
 - ◆ Verwendung von transaktionalen Schritten
 - ◆ Verwendung von einfacher flexibler Kompensationslogik
 - ◆ Kombination von SOA und BPM



SOA – aber wie? – Inge Hanschke – 21. Juli 2005
© iteratec GmbH iterative Softwaretechnologien, München, Frankfurt, Hamburg

Seite 24

Erfolgsfaktoren

- Basis für Erfolg schaffen
 - Unterstützung von der Unternehmensführung („Awareness“)
 - Budgets,
 - Pilot Projekt
 - Organisation: SOA Team und SOA Board
 - Prozesse, Strukturen und Standards
- Quick Wins
 - Nutzen für hohe Anfangsinvestitionen (SOA Infrastruktur)
- Substanzielle Änderungen: „Change“
- Business Involvement
 - Service-Herleitung aus den Prozessen
- Kommunikation! („Evangelize“)



Sicherheit und Web Service Architekturen

Erfahrungen mit Security Standards für Web Services



Sicherheit und Web Service Architekturen

Erfahrungen mit Security Standards für Web Services

M. Breu

Arctis Softwaretechnologie

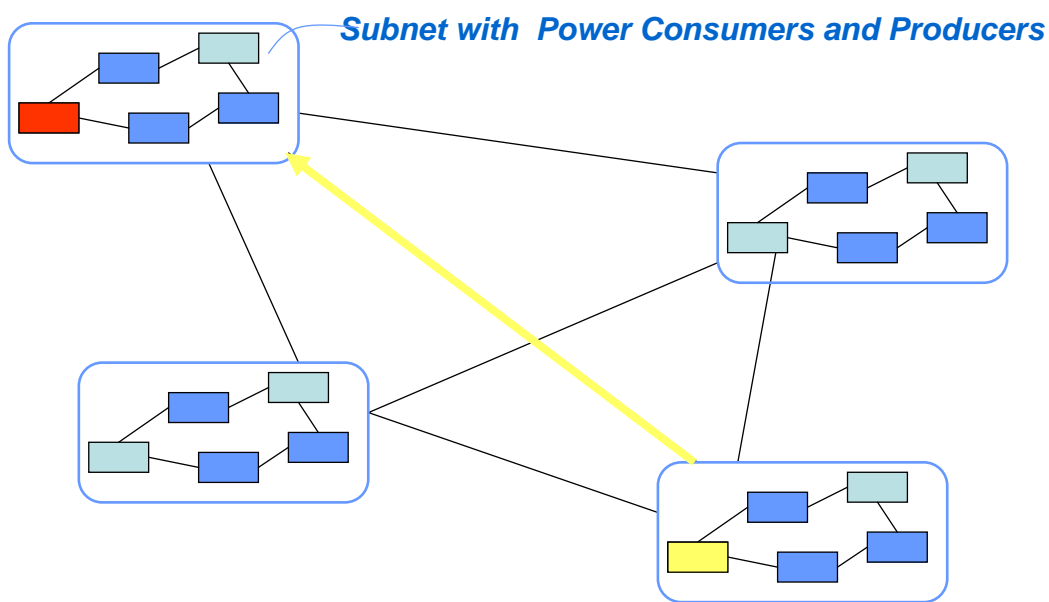


Content

- ◆ Motivation
- ◆ Model Centric Specification of Security and Workflows
- ◆ Standards and Components
- ◆ Target Architecture
- ◆ Conclusion



Example: European Power Distribution Market



Compensation of power shortage or overproduction between sub networks:
Upcoming European power market regulations demand facilities to awarding of power exchange contracts in tranches of 5 MW in 7.5 minutes



Motivation

◆ B2B and eGovernment Processes require

- ◇ the coupling of internal work flows with external partners
- ◇ adequate security management

◆ Technology of Choice: **Web Service centric architectures**

- ◇ wrapping internal functionality
- ◇ build on standards as SOAP, WSDL, WS-Security, XACML

◆ **But:**

- ◇ no guidance for the (application-level) integration of security
- ◇ web service security standards are emerging, but are technology oriented and highly complex

◆ **Resulting in ad-hoc solutions**

- ◇ costly, complex
- ◇ security requirements hard to maintain

Model Driven Security Architecture (1)

◆ **Specify Workflows and Security Requirements on Domain Level**

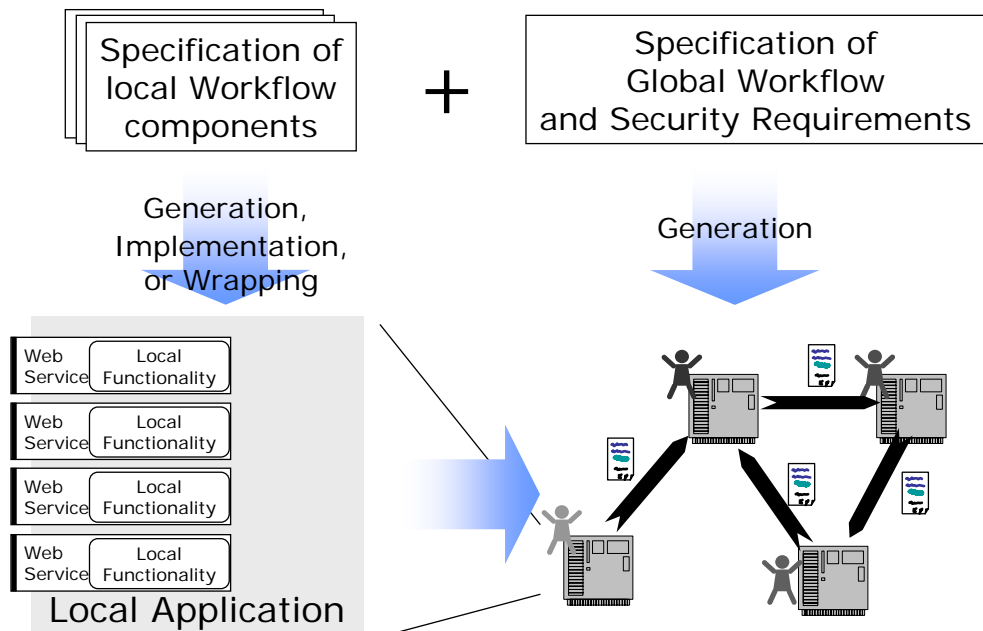
- ◇ globally (between various business partners)
- ◇ locally (internally to a business partner)



◆ **Derive components for a (given) Web Service centric Target Architecture**

- ◇ encapsulating the intricacy of the underlying technology

Model Driven Security Architecture (2)

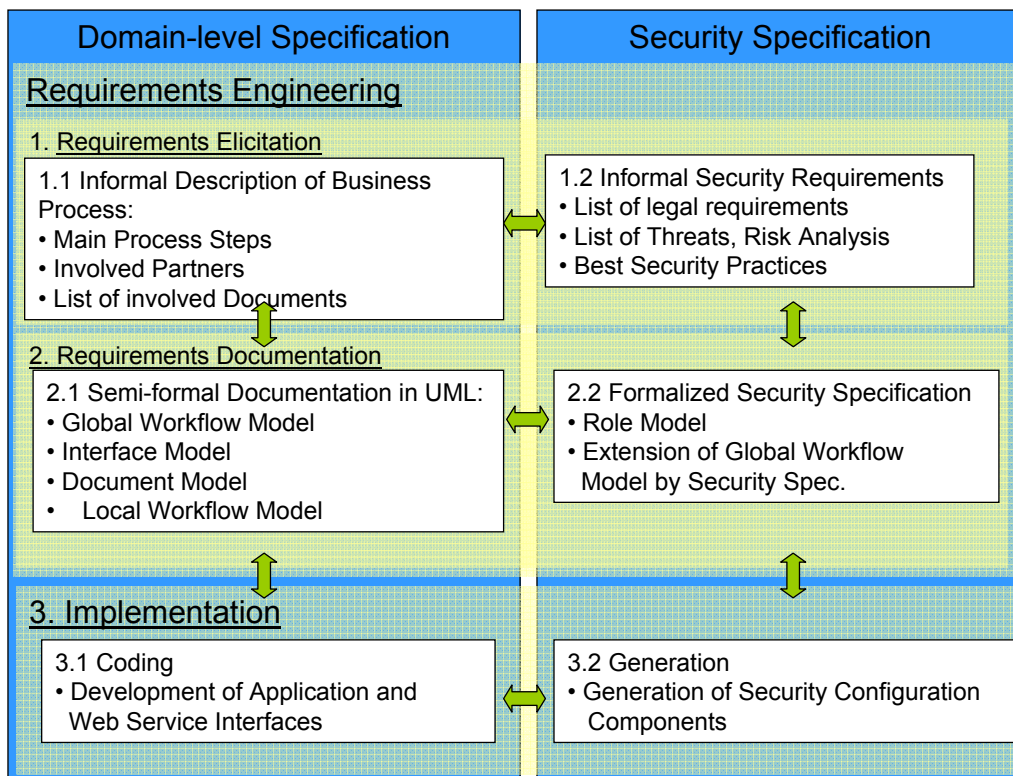


Enterprise application integration (EAI) in the large

Quality Engineering Presentation

7
22.07.2005

Process Model



Quality Engineering Presentation

8
22.07.2005

SECTINO Case Study

◆ Submission of Annual Statement and Municipal Tax Assessment

◆ Partners

- ◇ Corporation (tax payer)
- ◇ Tax Advisor
- ◇ Municipality

◆ Process Steps and Involved Document

1. The Corporation sends an *annual statement* to its Tax Advisor
2. The Tax Advisor reviews the statement for correctness and forwards it to the Municipality
3. The Municipality checks the annual statement and assesses the communal taxes
4. The Municipality forward a *tax notification* to the tax advisor, which reviews it for correctness and forwards it finally to the Corporation

Quality Engineering Presentation

9

22.07.2005

SECTINO Case Study: Security Requirements

◆ Examples

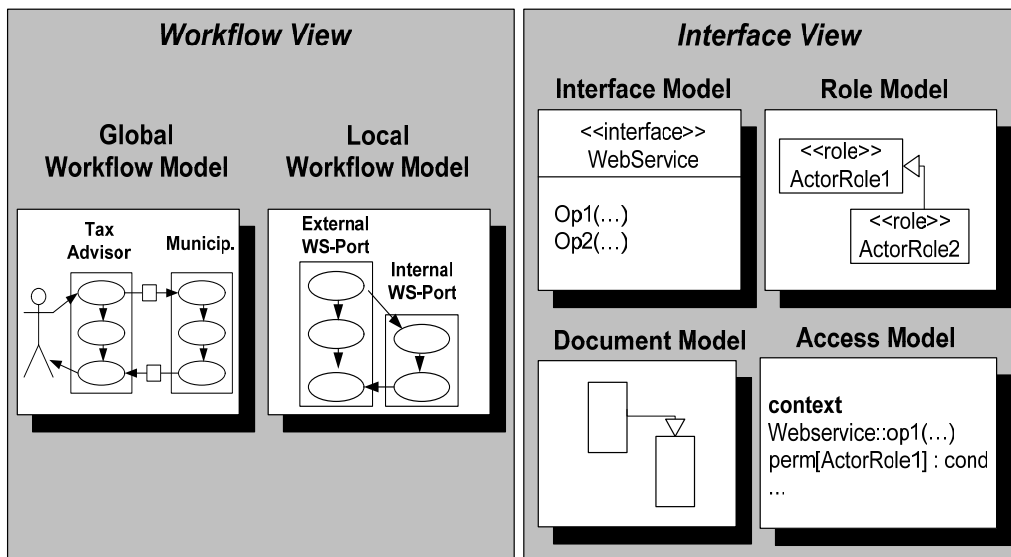
- ◇ All exchanged documents have to be signed by sending party
- ◇ The annual Income and the clientId of the annual statement and the tax notification are confidential, and are only be read by the Municipality
- ◇ The receipt of the annual statement and the notification must not be repudiable

Quality Engineering Presentation

10

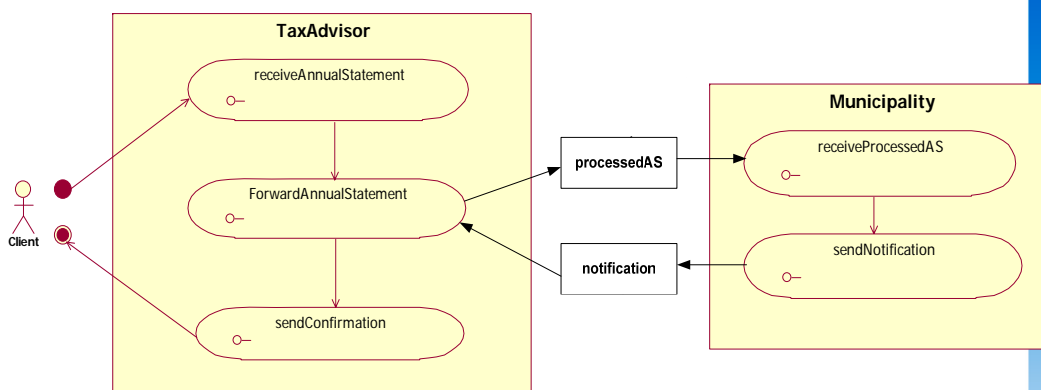
22.07.2005

The semi-formal documentation in UML



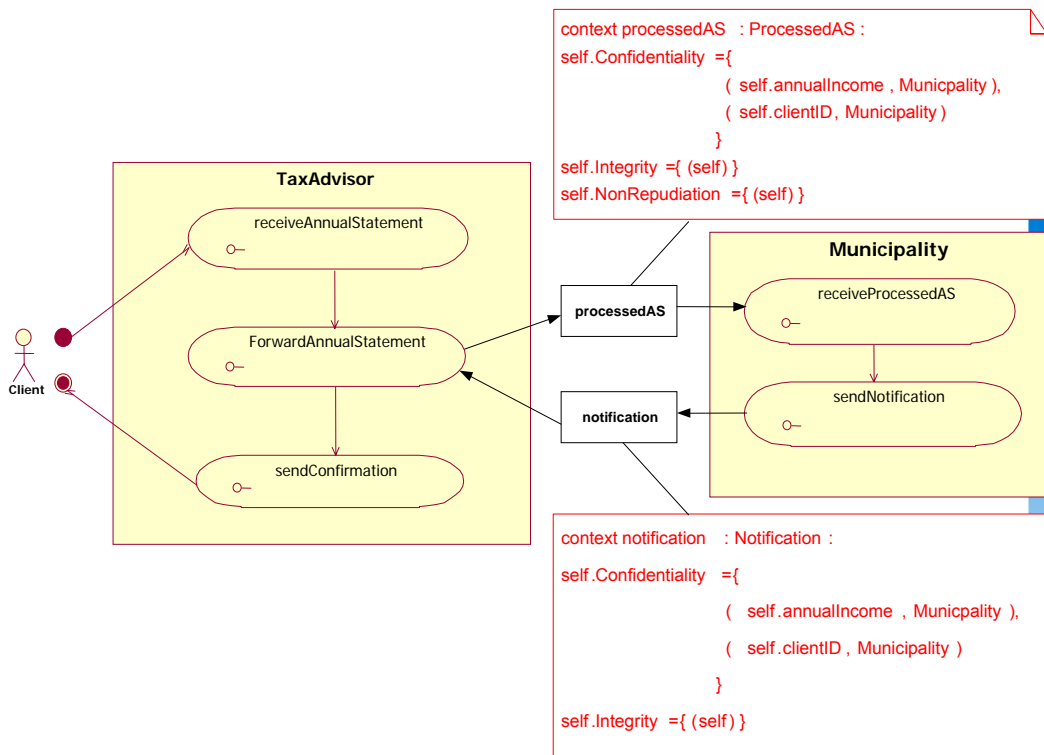
11

Global Workflow with Security Requirements



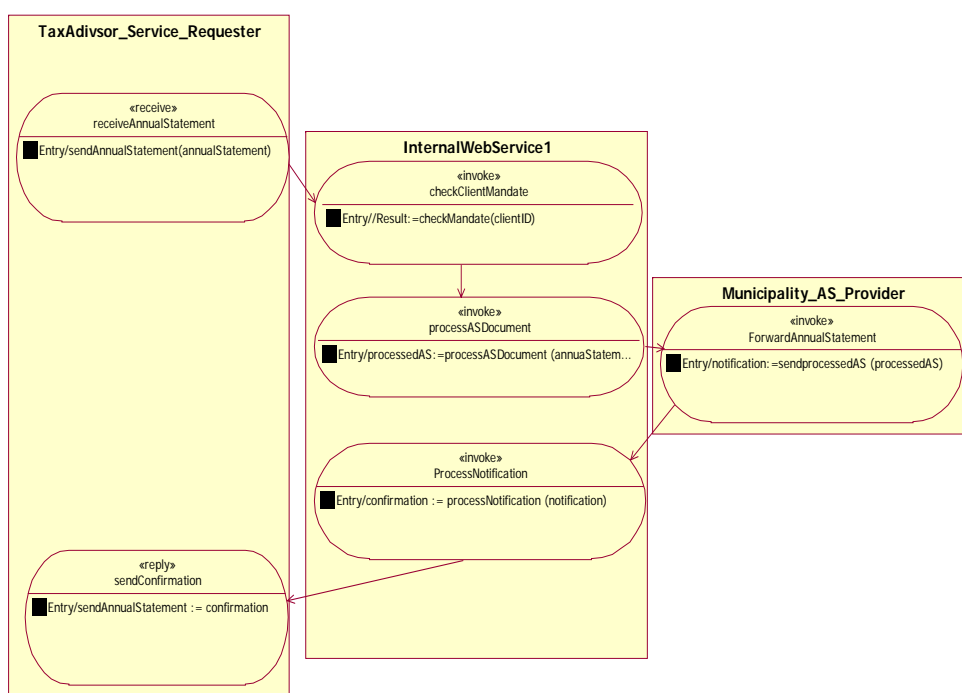
12

Global Workflow with Security Requirements



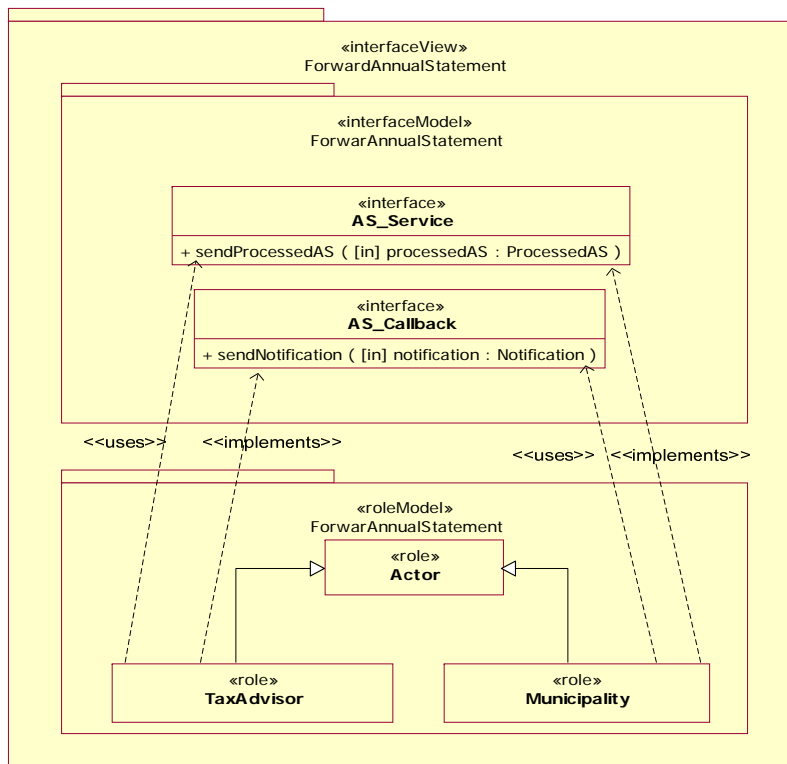
13

Local Workflow Model



14

Interface Model



15

22.07.2005

Access Model

◇ Access conditions can be specified

- ◇ in the interface model
- ◇ for individual roles
- ◇ using OCL (Object Constrained Language)

◇ Example

- ◇ A Tax Advisor can only submit a statement, if he is registered as a tax advisor for this company

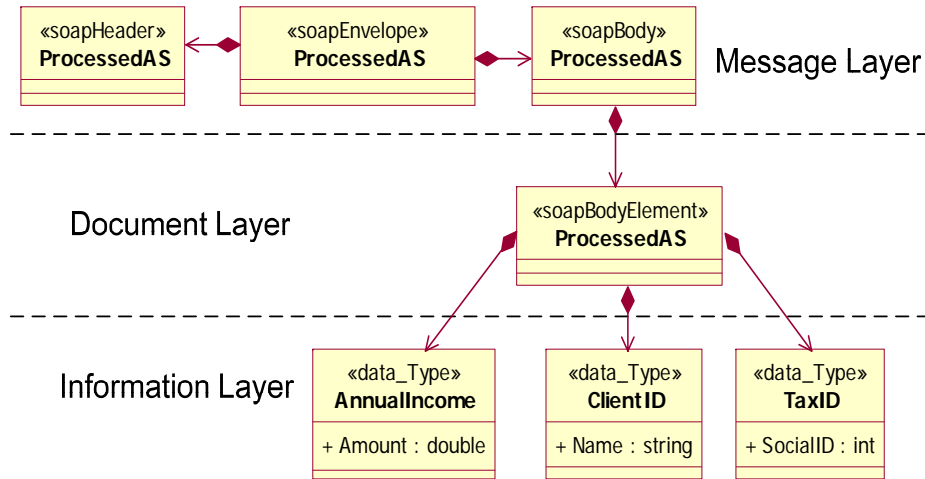
```

context AS_Service ::
    sendProcessedAS (processedAS: ProcessedAS)
perm [TaxAdvisor]:
    subject.map(TaxAdvisor).
        hasClient(processedAS.clientID)
perm [Company]:
    true
    
```

16

22.07.2005

Document Model



17

Quality Engineering Presentation

22.07.2005

Requirements for Target Architecture

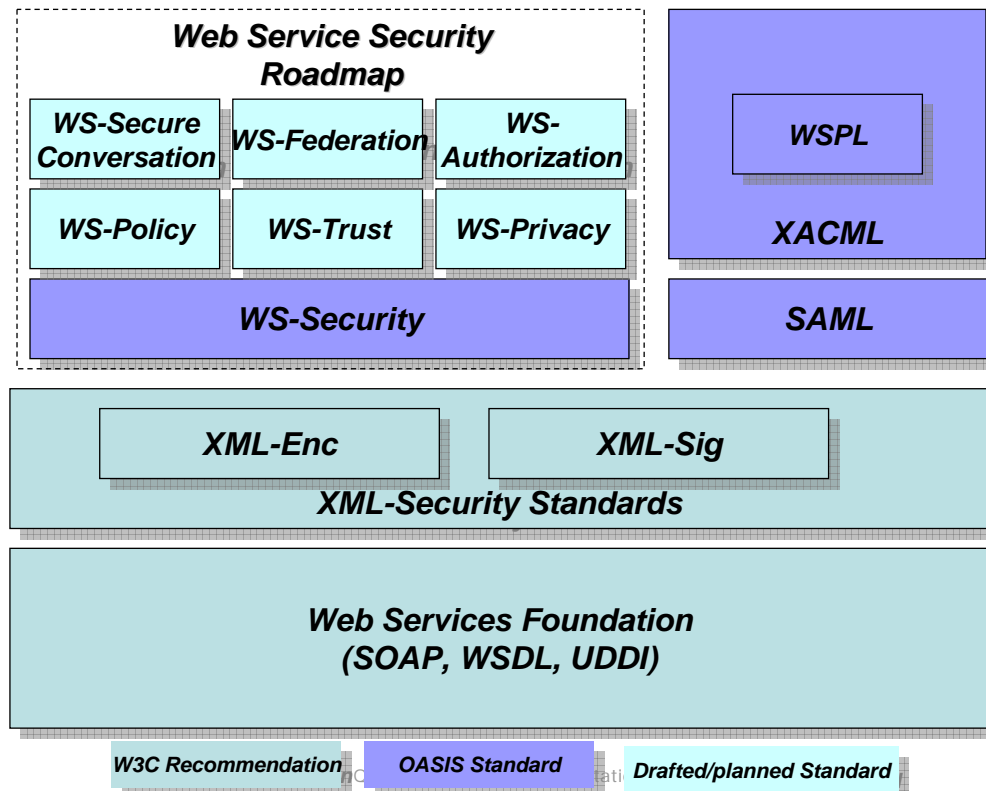
- ◇ **Built around core of atomic web services**
- ◇ **Support for Digital Security**
 - ◇ Digital Signature, Encryption
- ◇ **Security implementation by configuration**
- ◇ **Follows XACML workflow architecture**
 - ◇ Policy Enforcement Point (PEP)
 - ◇ Policy Decision Point (PDP)
- ◇ **External Services**
 - ◇ Public Key Infrastructure
 - ◇ Logging Unit (Non-repudiation)

18

Quality Engineering Presentation

22.07.2005

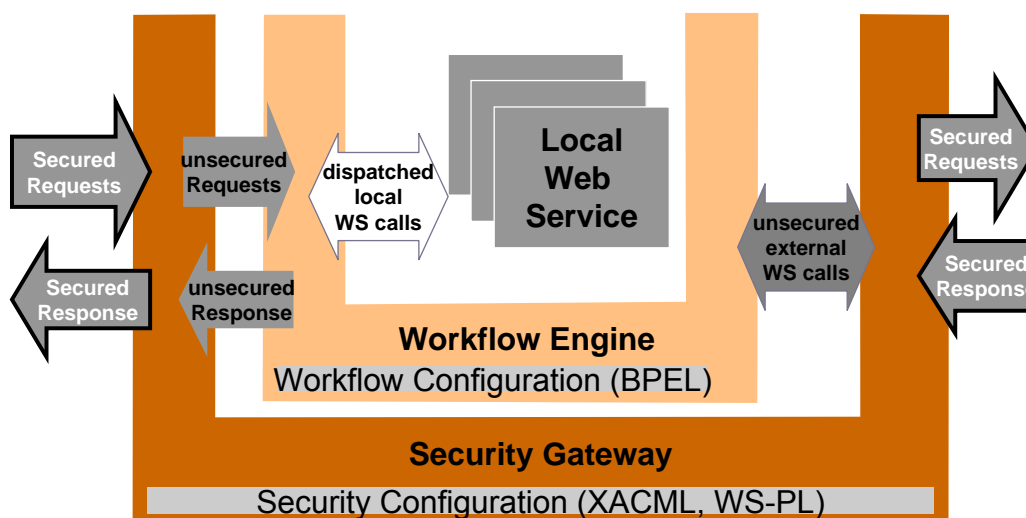
Web Service Security and related Standards



19

22.07.2005

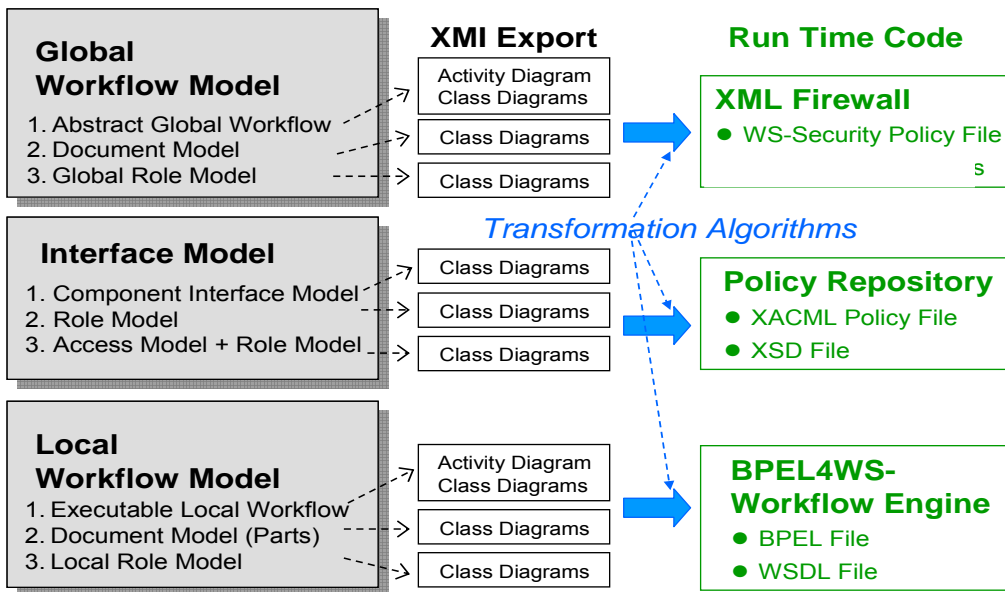
Schematic Target Architecture



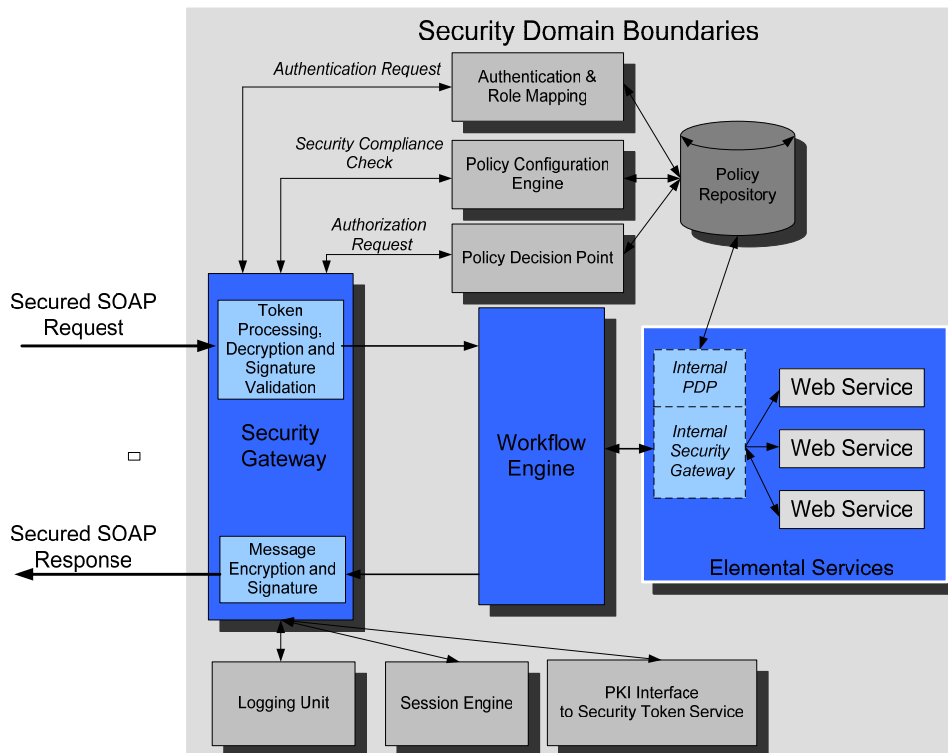
20

22.07.2005

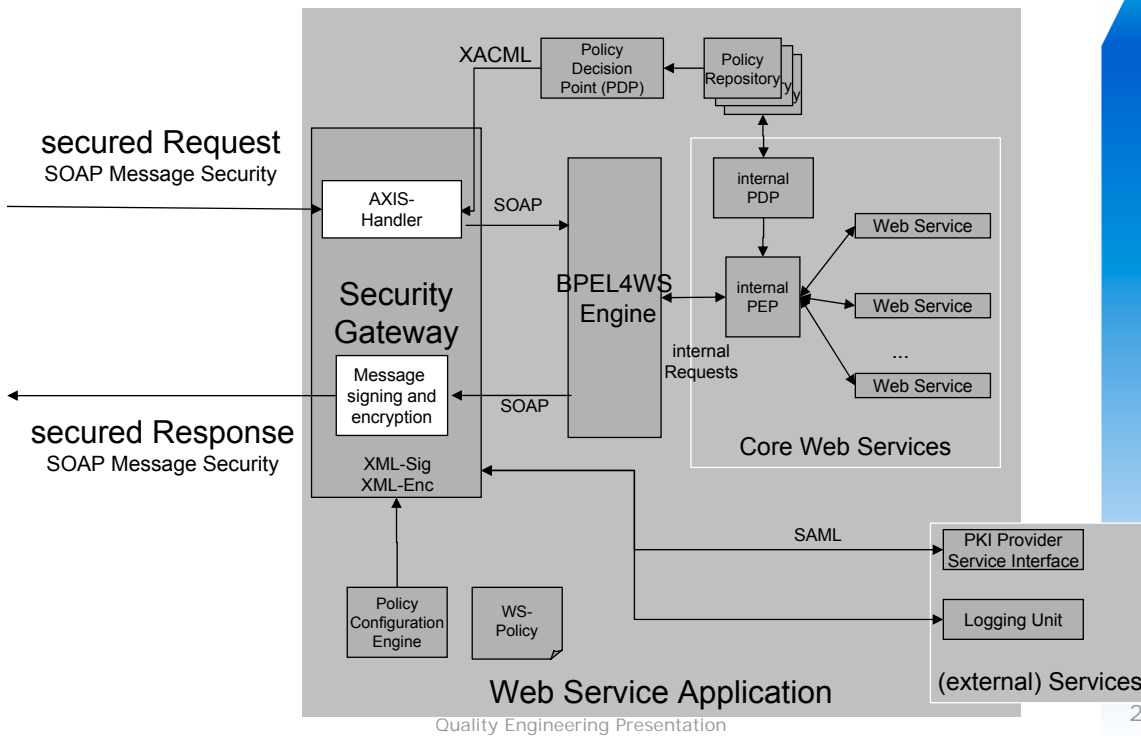
Transformation of Models into Configuration of the Target Architecture



Target Architecture Components



Target Architecture Details: Employed Standards



23

22.07.2005

Summary

- ◆ Modern B2B processes must consider security as a critical success factor
- ◆ Trend is from security islands to a holistic security approach (End-to-End security)
- ◆ The development of secure B2B processes is a complex engineering process
 - ◇ Guidance needed
 - ◇ Infrastructure Support needed

24

Quality Engineering Presentation

22.07.2005



Konzeption einer serviceorientierten Veranstalter-Plattform für einen Reiseveranstalter

München, 21.07.2004

Christian Wied

beck et al.
projects

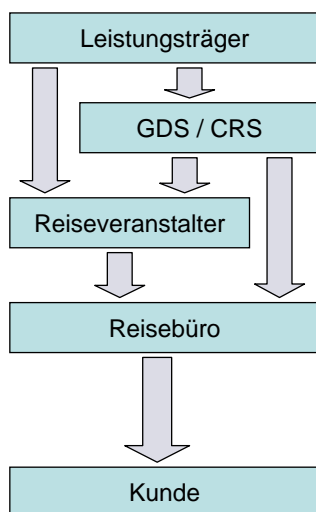
beck et al. projects

Theresienhöhe 13, 80339 München, T 089/54 42 53-0, F 089/54 42 53-99, info@bea.de, www.bea.de

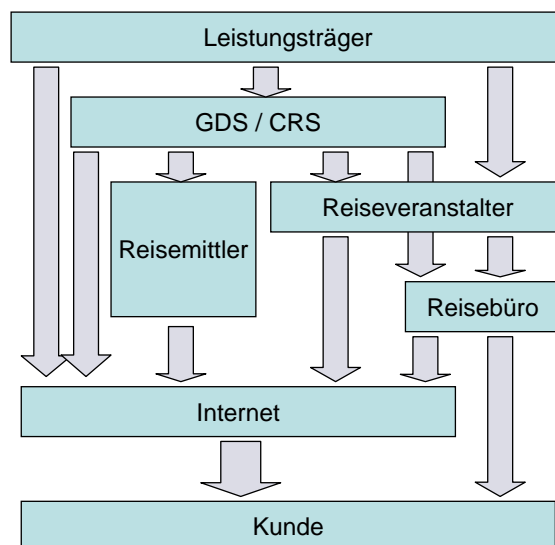


Pauschalreisen gestern und heute: Die Reiseveranstalter bekommen neue Konkurrenz.

Gestern:



Heute:



beck et al.
projects



Der Online-Vertrieb von Reisen ist für klassische Veranstalter herausfordernd.

- **Dynamische Produkte**
Anstelle des halbjährlich erstellten Katalogs sollen Produkte zukünftig jederzeit in das Angebot aufgenommen werden können.
- **Produktvielfalt durch externe Läger**
Zur Vergrößerung des Produkt-Portfolios sollen auch Leistungen von externen Partnern in das Angebot aufgenommen werden können.
- **Dynamic Packaging**
Anstelle der manuellen Erstellung der Pauschalreise sollen Angebote in Zukunft erst zum Zeitpunkt der Kundenanfrage paketiert werden.
- **Dynamic Pricing**
Der Preis der Pauschalreise soll in Zukunft dynamisch (z.B. anhand der prognostizierten Auslastung) zum Zeitpunkt der Kundenanfrage berechnet werden.
- **Differenzierung durch mehrere Marken**
Wie im Kataloggeschäft sollen auch im Internet Produkte unter verschiedenen Marken angeboten werden.



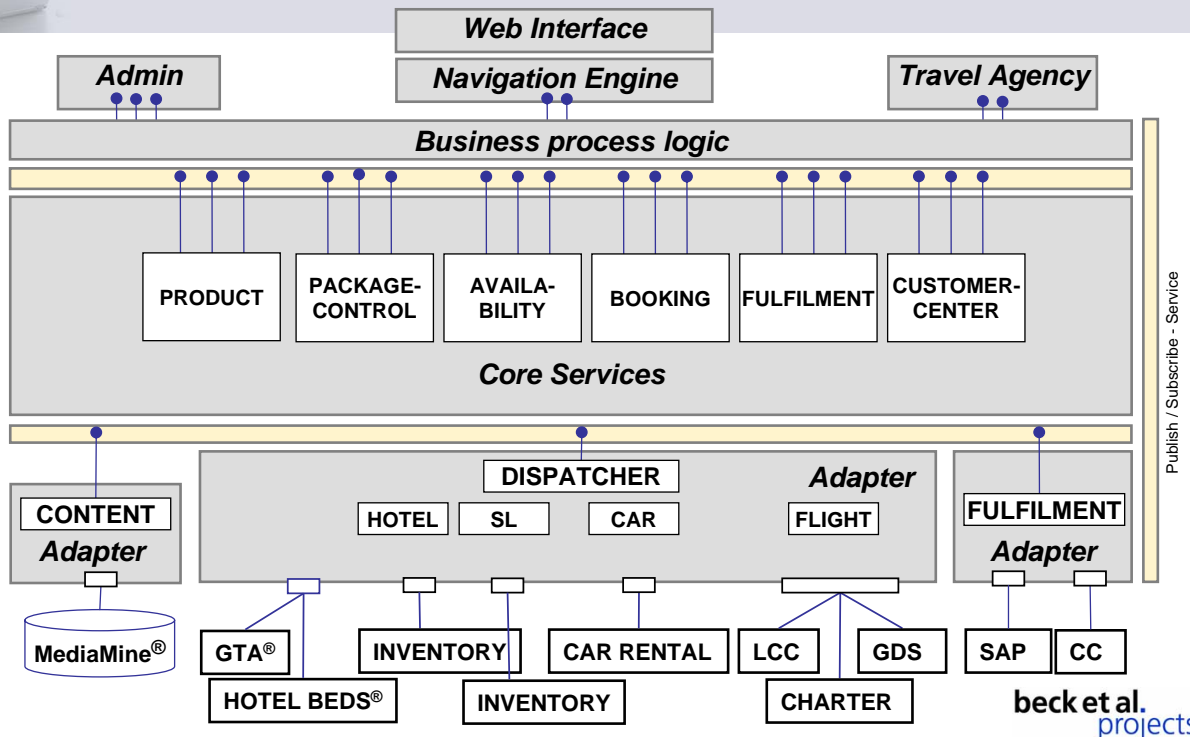
Hierfür muss eine neue und flexible IT-Lösung konzipiert werden!

beck et al.
projects

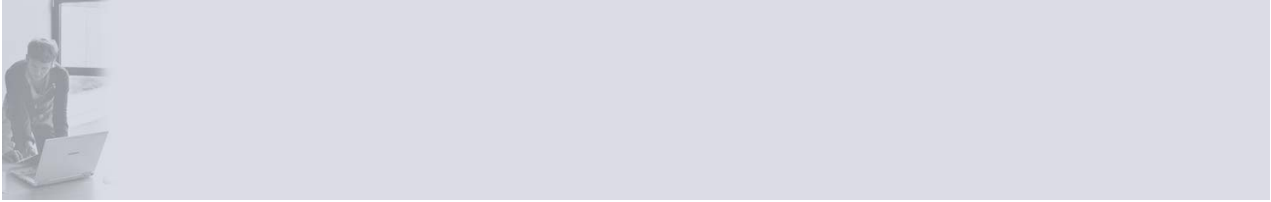
3 | Christian Wied / 21.07.2005



4-schichtige SOA!



4 | Christian Wied / 21.07.2005



Herzlichen Dank für Ihre Aufmerksamkeit!

Christian Wied Dipl.-Informatiker	beck et al. projects beck et al projects GmbH Theresienhöhe 13 D-80339 München Tel +49 / 89 / 5442533-84 Fax +49 / 89 / 5442533-99 www.bea.de christian.wied@bea.de
---	--

beck et al.
projects