

Serviceorientierte Architektur in die Realität umgesetzt

Andres Koch
dipl. El. Ing. HTL / M. Math
Object Engineering GmbH, Uitikon-Waldegg, Schweiz
Email: akoch@objeng.ch
www.objeng.ch

Zusammenfassung

Service-Orientierte Architektur (SOA) ist keine Neuerfindung, sondern eine logische Weiterentwicklung bisheriger Versuche, die Komplexität der über Jahre entstandenen System-Umgebungen in den Griff zu bekommen. SOA ist auch kein Allerheilmittel, sondern sie ist die Grundlage für die langfristige Pflege der IT-Architektur eines Unternehmens. Auf dem Papier mag es einfach erscheinen, die Realisierung präsentiert sich wesentlich schwieriger, da es sowohl technische wie menschliche Hindernisse zu überwinden gilt.

Alles fordert eine fast "diktatorische" Haltung des SOA-Architekten, der mit Diplomatie, Organisations-Geschick und natürlich Technologie-Kenntnissen aus dem heutigen das zukünftig optimale System-Umfeld schaffen muss.

Technologisch müssen bestehende, im Einsatz stehende Systeme ins gesamte SOA-Umfeld schrittweise eingebunden und die Komplexität besser verwaltbar gemacht oder sogar reduziert werden.

Das Alles ist kein Kinderspiel und trotzdem ist es machbar. Bei allen Hindernissen darf man die Vision und die Lebenserwartung (5-15 Jahre) nicht aus den Augen verlieren. Die gewonnenen Vorteile zahlen sich teilweise nur indirekt und oft auch erst nach einer geraumen Zeit aus. Man darf sich aber nicht durch die technischen Hypes blenden lassen, sondern soll dem gewählten Architektur-Weg treu bleiben.

Dieser Vortrag soll einige Tips aus der Praxis vermitteln und Denkanstöße geben wie die Problematik im Unternehmen gelöst werden kann.

Literaturverzeichnis:

- Enterprise Service Bus, David A. Chappell, O'Reilly, 2004, ISBN-0-596-00675-6
- Enterprise SOA, Krafzig/Banke/Slama, Prentice Hall, ISBN 0-13-146575-9
- Effektive Software Architektur (2nd Edition), Gernot Starke, Hanser, ISBN 3-446-22846-2

Serviceorientierte Architektur in die Realität umgesetzt

Andres Koch
dipl. El. Ing. HTL / M. Math
Object Engineering GmbH, Uitikon-Waldegg, Schweiz
Email: akoch@objeng.ch
www.objeng.ch

Serviceorientierte Architektur in die Realität umgesetzt

- Motivation
- Definitionen
- Kernprobleme
- Architekturen-Vergleich
- Umsetzung
- Zusammenfassung

Motivation:

Ideale Software-Architektur

Erwartungen an eine ideale Software-Architektur

- Flexibilität:** Fähigkeit schnell an neue Anforderungen angepasst zu werden
- Robustheit:** Immer wieder ähnliche Konstruktionen wirken sich positiv auf die Qualität aus.
- Kapselung:** Durch Kapselung von logisch zusammengehörigen Applikations-Komponenten wird eine Software-Landschaft flexibler und besser kombinierbar.
- Lose Kopplung:** Dank loser Kopplung wird die Flexibilität weiter verbessert und es entsteht kein massiver Monolith.
- Kosten:** Durch wiederverwendbare Komponenten, Know-How (von Entwicklung bis zum Betrieb) und Konzepte können die Kosten längerfristig reduziert werden.
- Langlebigkeit:** Software ist heute in einem Unternehmen nicht mehr nur ein System sondern zum Teil über Dutzende wenn nicht Hunderte von Systeme und Komponenten verteilt. Eine Architektur alle 3-5 Jahre zu ändern und an neue Technologien anzupassen widerspricht dem gesunden Menschenverstand.

Unternehmens-Software-Architektur kann wie *Stadtplanung* betrachtet werden.

Motivation:

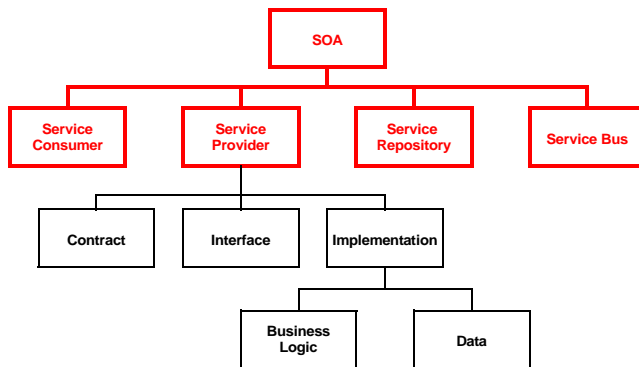
Gängige Software-Architekturen

<p>OMA: Object Management Architecture:</p>	<ul style="list-style-type: none"> • Oft einfache Client-Server Architekturen damit realisiert (CORBA) • Zweck der Referenz-Architektur war bereits Service-orientiert (Object Management Group). • Idee vom Service-Bus und von Services bereits geboren. • Geschäftsprozess aber in einzelnen Services (Applikation) • Synchrone Kommunikations-Topologie (RPC)
<p>EAI: Enterprise Application Integration</p>	<ul style="list-style-type: none"> • Systeme und Netze untereinander verbunden • Kein Konzept, einzig die Verbindung zählte. • <i>Spaghetti-Integration</i> resultierte (Punkt zu Punkt) • B2B als Hauptthema (Conversion, Connectivity)
<p>SOA: Service Oriented Architecture</p>	<ul style="list-style-type: none"> • Symmetrische Client-Server-Systeme auflösen • Peer-to-Peer Topologie • Asynchrone Kommunikation (Message-orientiert) • Lose Koppelung • Prozessablauf von Applikation getrennt. • Referenz-Architektur erst am Entstehen (OASIS?)

Definitionen:

SOA-Schlüsselkomponenten

Service Orientierte Architektur



Eine service-orientierte Architektur (SOA) ist eine Software-Architektur, die auf folgenden Grundkomponenten basiert:

- Service-Consumer
- Service-Provider
- Service-Repository
- Service-Bus basiert.

Quelle: [Krafzig et.al]

Definitionen:

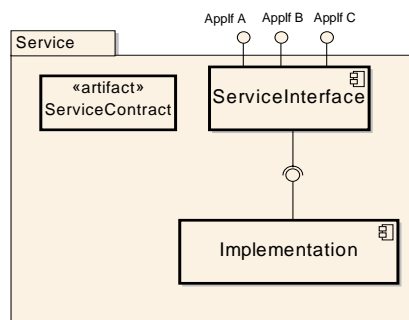
Service-Provider

Service-Provider

Ein Service besteht aus:

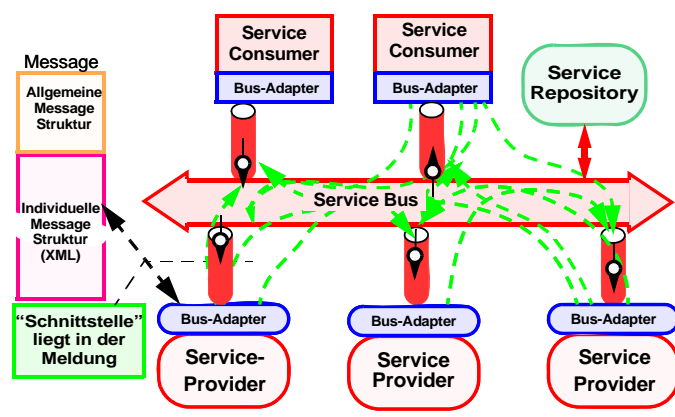
- einem Contract
- einer oder mehreren Schnittstellen
- einer Implementation

Quelle: [Krafzig et.al]



Definitionen:

Service Bus



- Der Service-Bus gehört nicht zur logischen Architektur-Sicht, sondern ist eine physikalische Infrastruktur-Komponente mit zentraler Bedeutung.
- Enthält Middleware und Kontroll-Funktionalität
- Bietet in der Regel viele Protokoll-Adapter an, welche als technische Services oder als Adapter angeschlossen werden (EAI-Teil)

- Erfüllt gewissermassen die Aufgabe einer *Telefon-Zentrale* für Services mit folgenden Eigenschaften: Durchgängigkeit (Pervasiveness), Stark verteilt (highly distributed), Routing, Sicherheit (Authentication, Authorisation), Vermittlung (Routing, Registry), Plattformübergreifend, Normierte Schnittstellen
- Wird heute durch sogenannte ESB *Enterprise Service Bus* realisiert.

Architektur-Vergleich:

OMA & SOA

Thema	Object Management Architecture	Service Oriented Architecture
Middleware	CORBA/IIOP/synchron/RPC-Style	noch proprietär(JMS)/asynchron/MOM-Style (Ausnahme WS*-Standards)
Technologie	robust, relativ starre Schnittstellen, generell schnell, standardisierte Technologie (IIOP), plattform- und sprachen-übergreifend	flexibel, stark interpretiert, genug schnell, einige Technologie-Standards, meist bestehende MOM-Basis. Interoperabilität je nach Hersteller gewährleistet. WebService-Technologie unterstützt.
Performance	sehr hoch	mittel, genügend (abhängig von MW)
Paradigma	Client/Server (einfachste)	Peer to Peer (komplexer aber allgemeiner)
Flexibilität	relativ starr, mit gewissem Aufwand erreichbar.	sehr hoch, hat seinen Preis (Performance, Interpretationsaufwand)
Prozesse	typ. in der Applikation, muss von aussen unterstützt werden (Workflow-Engines)	nicht in Applikation, auf Orchestrierungs-Ebene (technisch wie Business). Im Grundkonzept (ESB) enthalten.
Implementation	benötigt ein gewisses Grund-Knowhow	Integration von Applikationen wird gut unterstützt, XML-Know-How unumgänglich

- Governance unbekannt oder übertrieben (Elfenbeinturm-Sicht)
- Geschäftsprozesse nicht im Griff, Kluft zwischen Business und IT
- Menschliche Unflexibilität, Hang zum Alten und fehlende Bereitschaft etwas zu wagen und Neues zu lernen.
- Zu kurzfristiges Denken, kein realistischer SOA-Architektur-Horizont (5-10 Jahre).
- Schlechte Sichtbarkeit des ROI, dadurch schlechte Argumente für entsprechende Anfangsinvestitionen.
- Bestehende System-Monolithen und komplexe, enggekoppelte Datenmodelle lassen sich schlecht entflechten.
- Komplexität wird verschoben, aber nicht verringert (Zentraler Admin-Stelle).
- Zu stark der Technologie gehorchend, statt eine systematische Architektur zu verfolgen.
- Existierende Systemlandschaften mit hoher, unübersichtlicher Komplexität.
- Oft "Silo"-Architekturen vorhanden und weiterhin gepflegt (1 Client, 1 Service)
- Wenig oder keine Architektur oder/und Schnittstellen-Systematik

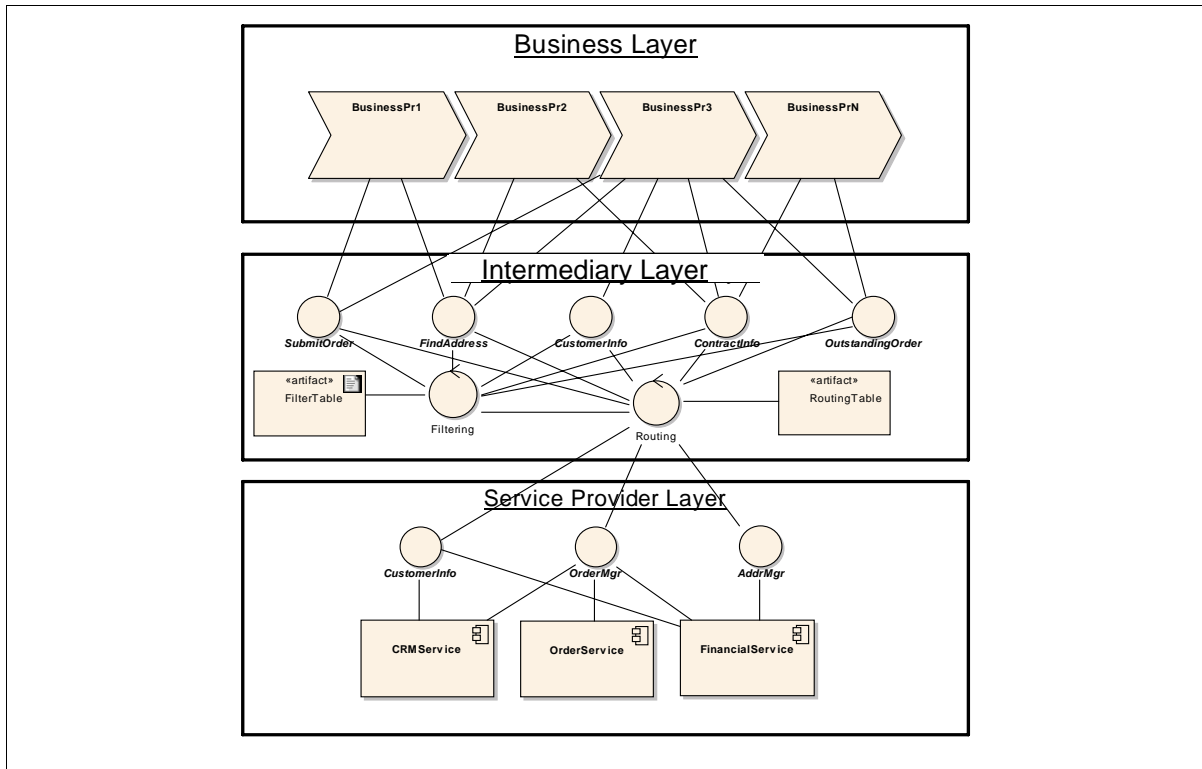
Es gibt Abstufungen der SOA-Reife eines Unternehmens:

- **Service-Provider Fassade:** Services entstehen anforderungsorientiert, werden u.U. noch direkt oder via Service-Bus angesprochen. Anforderungsorientiert
- **Ablauf-Prozesse** (technisch) auf Ebene des Service-Bus definiert und kombiniert mit Transformations-Prozessen. Integration mit Legacy-Systemen über Service-Bus integriert.
- **Unternehmens-Geschäftsprozesse** ermittelt, definiert und u.U. überarbeitet (Redesign) und werden schrittweise über einen Workflow-Service realisiert.
- **Unternehmensübergreifende Geschäftsprozesse** ermittelt und definiert, mit externen Partnern abgesprochen und realisiert.

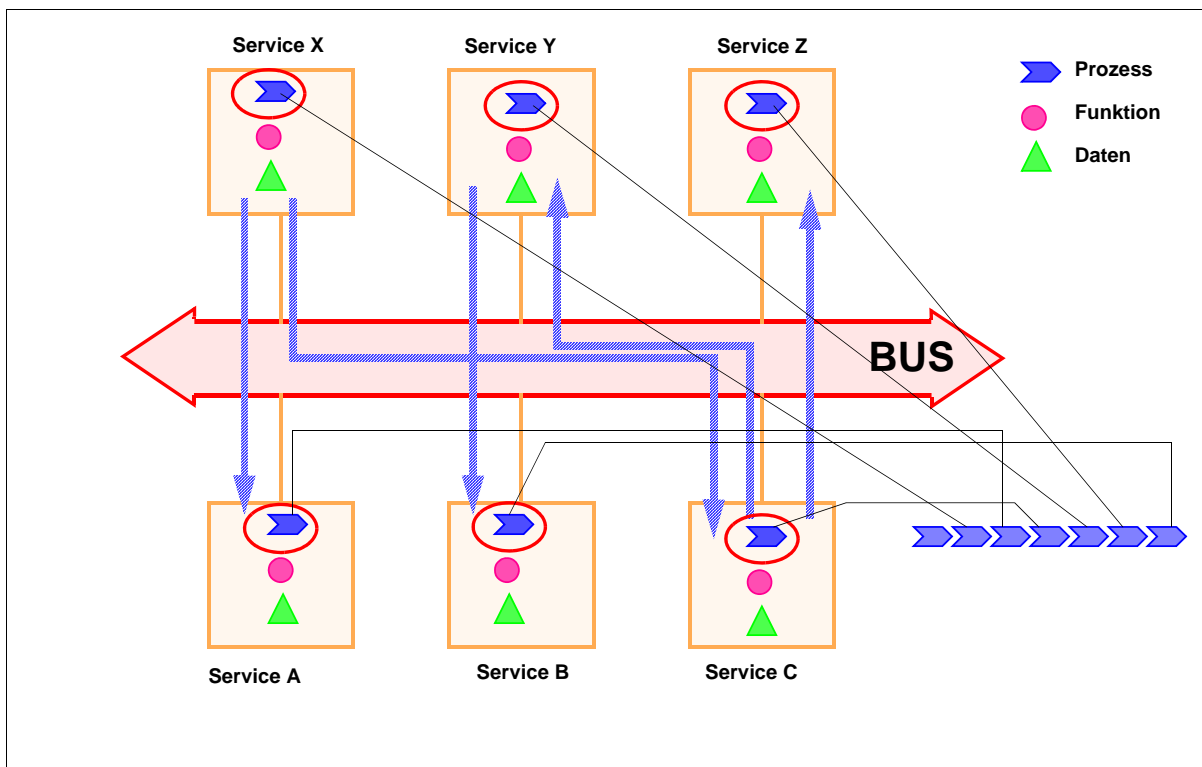
Diese Reife kann sowohl horizontal oder in vertikalen Durchstichen umgesetzt werden.

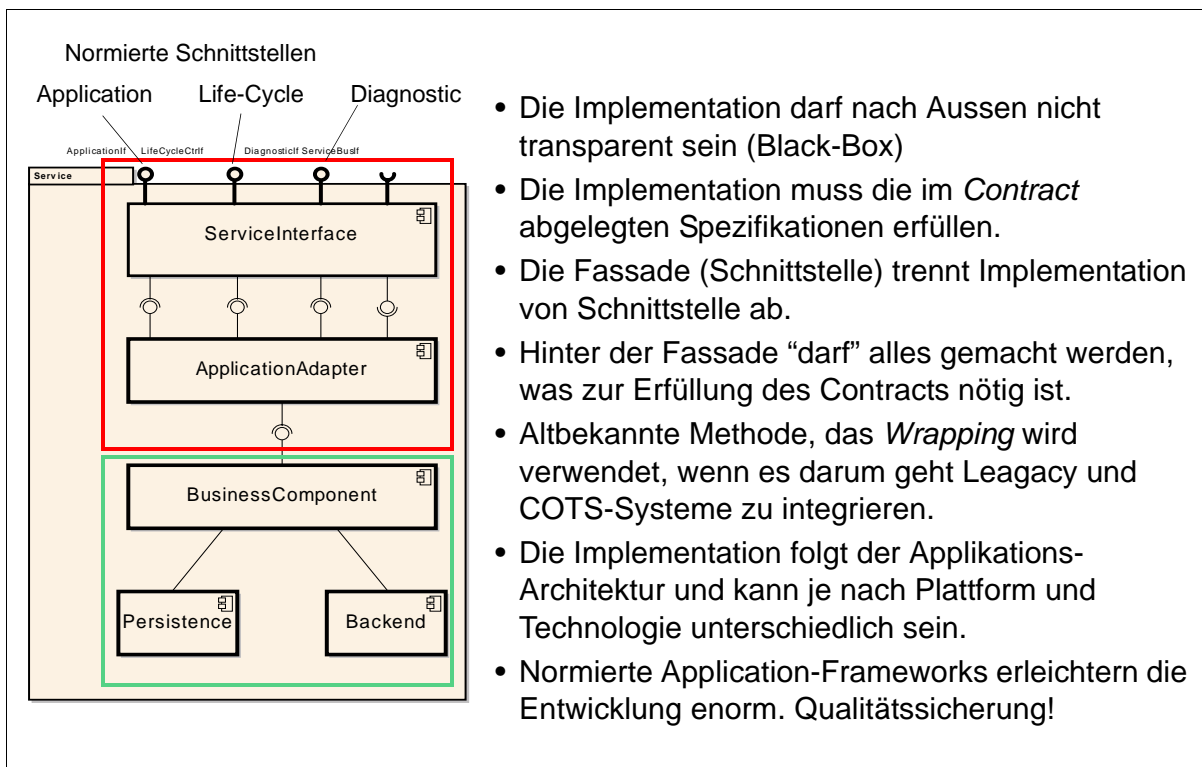
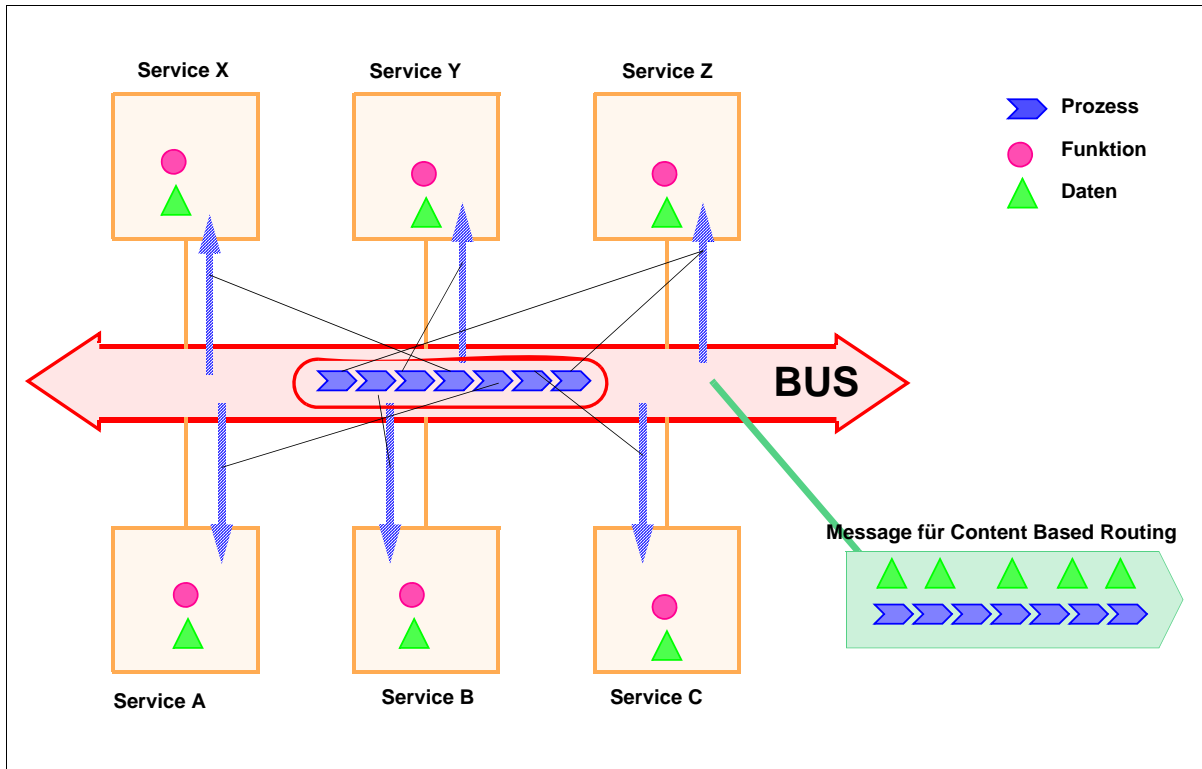
Umsetzung

SOA-Layer-Architektur



Bisher: Prozess in Applikationen eingebettet

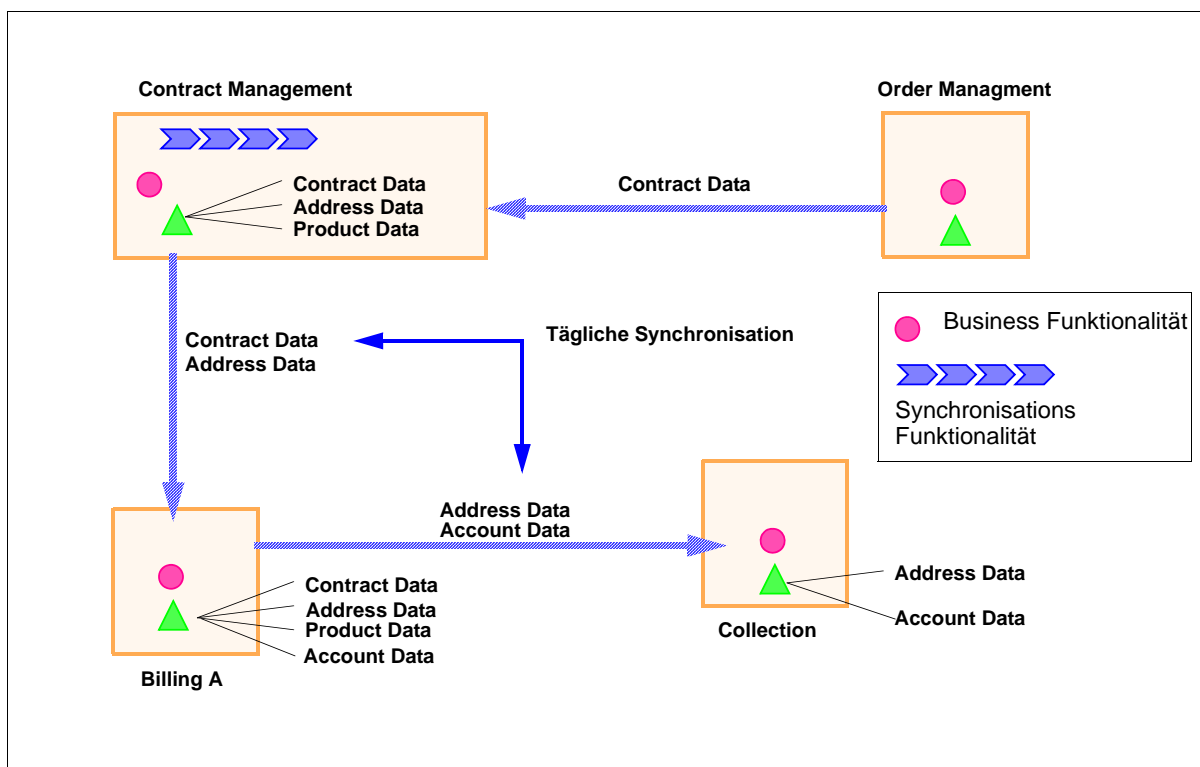




Schnittstellen-Design das A&O

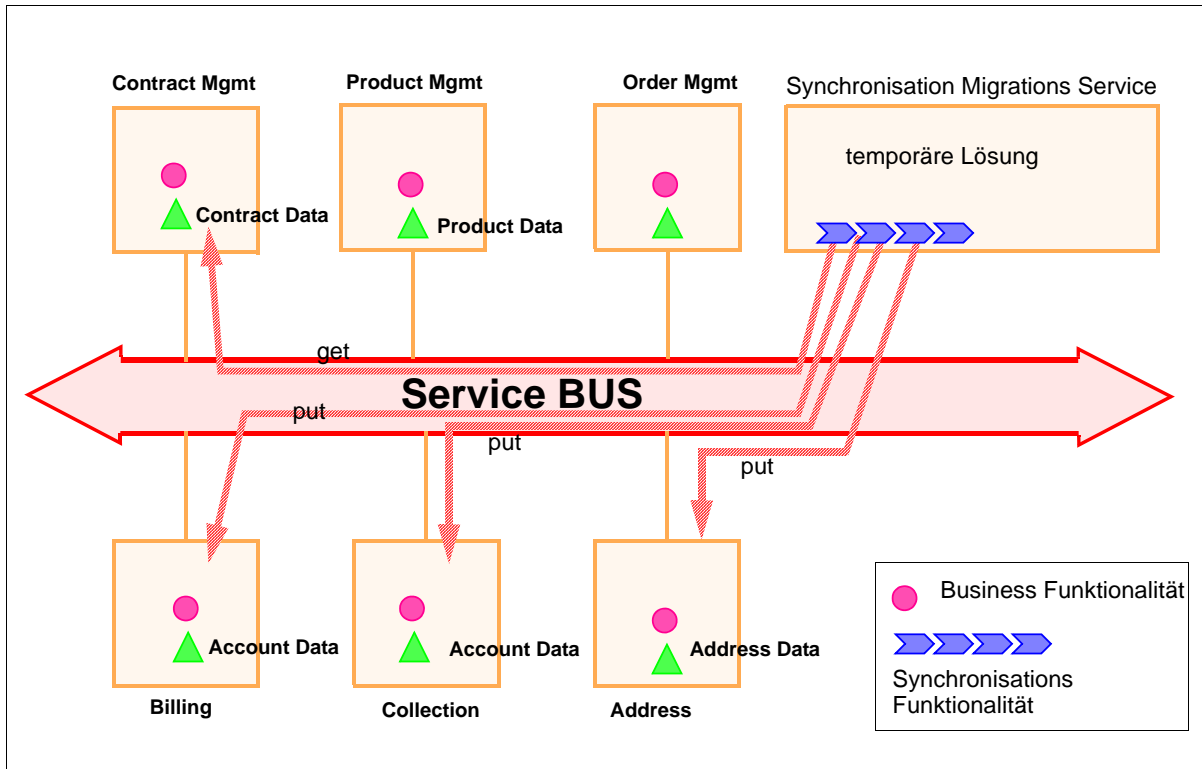
- Schnittstellen sollten aufgrund der **Anforderungen** entworfen werden.
- Den Service-Schnittstellen muss in der Entwurfsphase genügend **Aufmerksamkeit** und **Sorgfalt** gewidmet werden.
- Es ist vertretbar, wenn eine Schnittstelle während der Entwurfsphase ändert. Dies nach deren Inbetriebnahme zu tun, kann sich fatal auswirken.
- Schnittstellen dürfen nur erweitert werden, wenn eine **Aufwärtskompatibilität** sichergestellt werden muss.
- Qualitäten wie Performance und Flexibilität sind gegenläufig.
- Message-Modelle erstellen, damit normalisierte Meldungen entstehen.
- Mit einer Interface Definition Sprache (WSDL, IDL) können Schnittstellen definiert werden, die unabhängig von der in der Implementation verwendeten Programmier-Sprache ist.
- Auch aus UML-Modellen können Schnittstellen-Definitionen generiert werden.
- Technologien und Tools ersetzen das Design **NICHT!**
- Middleware-Protokoll ist weitgehend normiert (SOAP, IIOP, JMS u.a.)

Wichtig: Das Schnittstellen-Design nicht die Technologie ist entscheidend!



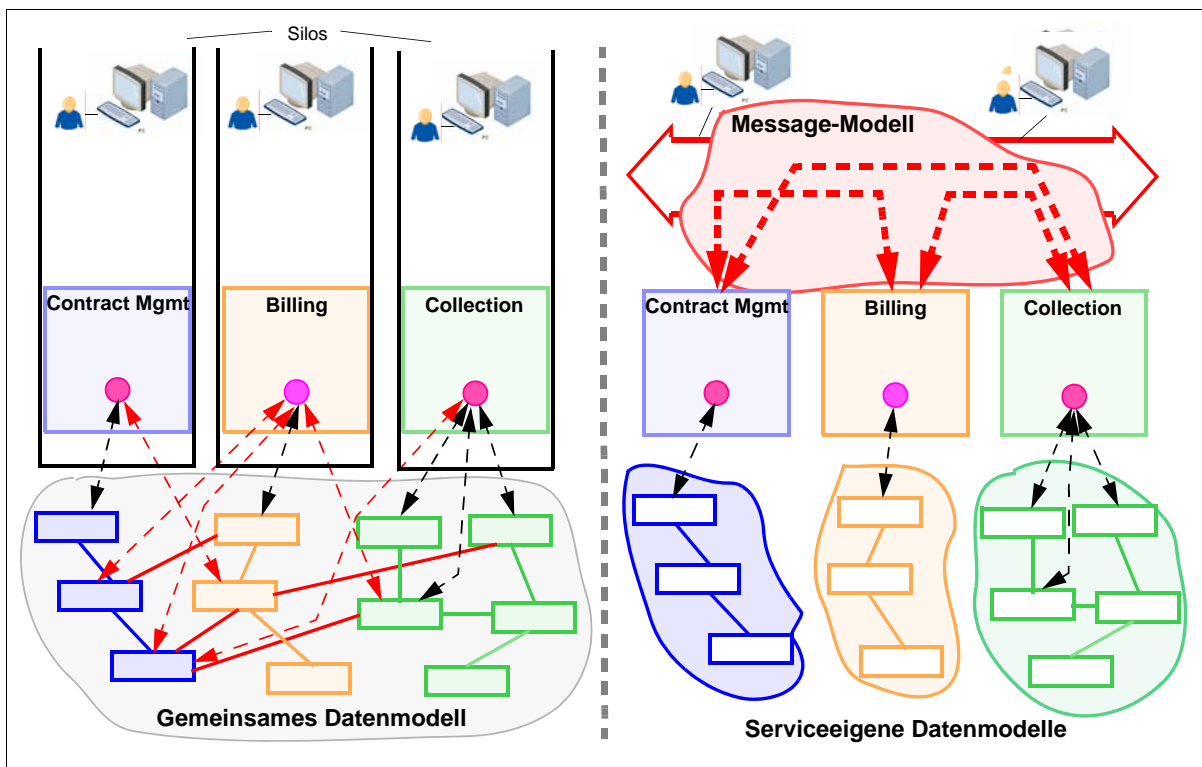
Umsetzung

Migrations-Beispiel



Umsetzung

Business Object Modell in SOA



- SOA in einem bestehenden Unternehmen einführen, erfordert organisatorisches und technologisches Geschick.
- Das obere Management muss hinter dem Unterfangen stehen und den nötigen Durchsetzung-Druck ausüben.
- Zu viele Köche verderben...!
- Senior-Architekt(in) durch Spezialisten für Teilgebiete unterstützen.
- Prozesse müssen für das Erstellen wie das Ändern von publizierten Services aufgesetzt werden. -> SOA-Board.
- Bestehende Teams, welche heute die Kernsysteme weiterentwickeln und pflegen, werden mittelfristig in Service-Teams umgestellt.
- 2er-Teams von Business-Domain-Experten und IT-Fachleuten sollten für die Umsetzung der Business-Prozesse gebildet werden.
- Business-Prozesse sind die grosse Herausforderung.
- Change-Management ist nicht trivial.
- Während dem die aktuellen Projekte weiter laufen, muss SOA aufgesetzt werden. Dies geht nicht von heute auf morgen.
- Wohlüberlegte Schritte führen früher zum Ziel.

Organisation, Planung vor Technologie

- Kein Big-Bang-Ansatz wählen, sondern Schritt für Schritt vorgehen. Dazu ist SOA prädestiniert.
- Typisch hat man eine bestehende Umgebung, die es einzubinden gilt.
- Lebenserwartung (Vision) einer SOA sollte min. 10-15 Jahre sein.
- Organisatorische Massnahmen müssen parallel mitlaufen (ähnlich Klassenmanagement zu OO-Zeiten).
- SOA-Board für das Validieren und Zulassen von neuen SOA-Services.
- Massenverarbeitungsprozesse können via SOA-Services angestossen werden. Bei grossen Datenmengen und Durchsatzraten, müssen Datenkopien temporär erstellt und verwendet werden. Technischer Aspekt.
- SOA-Realisierung läuft parallel zum Tagesgeschäft und den laufenden Entwicklungs-Releases.
- Frühzeitig die Projekte informiert halten, Architektur im Elfenbeinturm ist auch bei SOA fast ein Garant für Misserfolg.
- Technologie ist ein untergeordnetes Thema.

Schlussfolgerung

- SOA hat in erster Linie mit IT-Architektur und Organisation und erst an zweiter Stelle mit Technologie zu tun.
- Architektur aus einer Hand erweist sich als vorteilhaft
- Gutes Teamwork und gute Kommunikation auf menschlicher Ebene ist für den Erfolg von SOA-Projekten unabdingbar. Bremsende Elemente frühzeitig ausscheiden.
- Technologie ersetzt auch bei SOA weder den guten Projektleiter noch den kreativen Software Ingenieur.
- Das Thema automatisierter Business-Prozess ist vielerorts noch nicht wirklich konfrontiert. Die Verständnislücke zwischen Business und IT-Technologen ist nach wie vor da.
- Neues Paradigma kennen lernen und ausnutzen (z.B. Messaging)
- SOA braucht Mut, Kreativität aber auch Pragmatik, ohne dabei Systematik und Professionalismus zu vergessen.
- IT-Architektur ist wie Qualitätssicherung ein fortwährender Prozess.
- SOA für 2-3 Jahre lohnt sich nicht, die Perspektive der Lebensdauer muss bei 5-15 Jahre liegen.
- In kleinen Schritten stetig vorwärts schreiten, führt sicher zum Ziel.

Besten Dank für Ihre Aufmerksamkeit!

Fragen?