

Service-orientierte Software-Architekturen

Prof. Dr. U. Hoffmann
FH Wedel

Organisatorische Aspekte und
SOA im Unternehmenskontext

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Organisatorische Aspekte

Rollen und Organisationen
Finanzierungsmodelle

SOA im Unternehmenskontext

Architekturmodelle
Auswirkungen auf Front- und Backends

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Wer entwirft eigentlich neue Services?

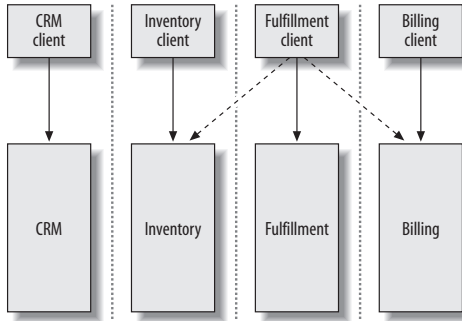
Diskrepanz zwischen

- ▶ verteilter IT (insbesondere SOA) und
- ▶ traditionellen Unternehmensstrukturen:
 - ▶ funktionale Aufteilung
 - ▶ statische Abteilungen und Unterabteilungen verantwortlich für die jeweiligen Systeme
 - ▶ Auftragsvergabe meist abteilungsbezogen
 - ▶ Projektgruppen werden häufig schnell zu Abteilungen

Conways Gesetz (Conways Law):

„Wenn Organisationen Systeme entwerfen, führt das zwangsläufig zu Systemen, die Kopien ihrer Kommunikationsstruktur darstellen.“

Melvin Conway, 1968



In der **Säulen–** oder **Silo–Struktur** traditioneller Unternehmens–IT muss ein interner oder externe Kunde typischerweise mit mehreren Systemen zusammenarbeiten.

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Einführung verteilter Geschäftsprozesse führt zu

- ▶ verteilter Planung
- ▶ verteiltem Design
- ▶ verteilter Umsetzung
- ▶ verteiltem Betrieb
- ▶ verteilter Problembehandlung

⇒ Ablösung des Abteilungsdenken
(*Fürstentum*–Denken)

Herausforderung: Bei verteilter Entwicklung hat nicht mehr eine Abteilung die Verantwortung über einen ganzen Prozess.

Zusätzliche Organisationsstrukturen ergänzen Abteilungen:

- ▶ *Projektorientierte Organisationsstruktur* für die Entwicklung neuer verteilter Funktionalitäten
- ▶ *übergreifende, Querschnitts-Abteilungen* für die Pflege abteilungsübergreifender Funktionalitäten

Erstes Beispiel für Abteilungen dieser Art:

Fullfillment, Auftragsverwaltung

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Solution–Management ist eine Strategie zur Entwicklung verteilter Funktionalitäten.

Projektmanager

(Solution–Manager, Anforderungsmanager, Projektleiter)

- ▶ kümmert sich um die vollständige Verwaltung und Umsetzung einer neuen Funktionalität.
- ▶ bringt die Kompetenzen und Daten zusammen, um einen funktionierenden Geschäftsprozess ins Leben zu rufen.

1) Design auf hoher Ebene (*High Level Design, Solution–Design*)

- ▶ Welche existierenden Systeme sind betroffen?
- ▶ Welche Firmen, Standorte, Abteilungen, Teams müssen einbezogen werden?
- ▶ Welchen Anteil haben sie an der Lösung?
- ▶ Besonders grundlegend:
 - Welche Backend–Abteilungen spielen eine Rolle?
 - Pflege und Bereitstellung von Daten und Geschäftsregeln
- ▶ Grundlage für Aufwandsabschätzung und Wirtschaftlichkeitsrechnung

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

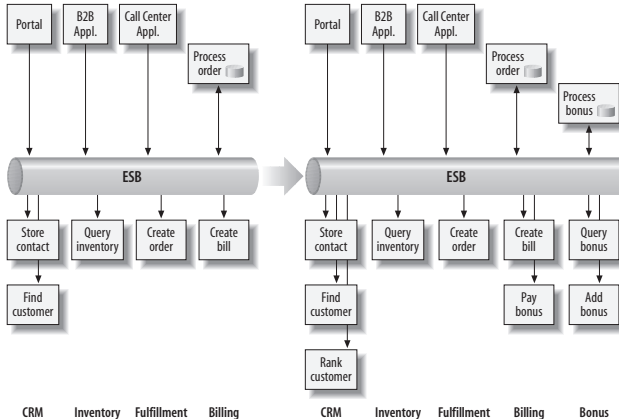
im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Beispiel: Einführung eines Bonus–Systems



Graphik: *SOA in Practive*, N. Josuttis

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Weitere Aufgaben des Solution–Managers:

2) Begleitung bei Detail–Entwurf

- ▶ Solution–Manager *treibt* das Projekt voran und koordiniert die Arbeiten der Beteiligten.

3) Begleitung bei der Realisierung (Implementierung, Integration) der Lösung

Solution–Manager stellt sicher, dass

- ▶ alle Beteiligten rechtzeitig ihren Anteil an der Lösung beitragen.
- ▶ Probleme rechtzeitig eskalieren.
- ▶ im Projekt durchgehend miteinander kommuniziert wird.

4) Projekt–Ende:

- ▶ Lösung geht als Teil des bestehenden Systems in Betrieb
- ▶ Verantwortung geht in die Abteilungen über

5) Problemanalyse bei konzeptuellen Problemen

- ▶ übergreifende Klärung des Problems
- ▶ ggf. Start eines neuen Projekts

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Zusammenarbeit zwischen den Beteiligten ist ein notwendiger Faktor für eine erfolgreiche SOA.

Fürstentümer oder Profit-Center

- ▶ Entscheidungen über Entwicklung werden eigenständig getroffen
- ▶ vollständige Kontrolle über Entwicklung
- ▶ eigener Nutzen vor Gesamtnutzen des Systems oder der Firma

vs. kooperative Abteilungen

- ▶ Kontrolle (teilweise) an Solution-Manager abgeben
- ▶ Abhängigkeit von anderen Abteilungen eingehen

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Beim Umstieg auf projekt-orientierten Entwurf können Abteilungen problematische Verhaltensweisen an den Tag legen.

mögliche Probleme

- ▶ *Geheime Tagesordnung*: Es wird ein anderes als das offizielle Ziel verfolgt.
- ▶ *Wer zuerst zuckt, verliert*: Probleme werden solange unter der Decke gehalten, bis klar ist, das Projekt-Probleme von anderen verursacht werden.
- ▶ *Schwarzer Peter*: Durch taktische Manöver wird versucht, die eigene Abteilung im besten Licht dastehen zu lassen und die Ursachen für eine Projektschiefelage anderen zuzuschieben.
- ▶ *Dienst nach Vorschrift*: Absprachen werden dem Buchstaben aber nicht dem Geist nach erfüllt.

SOA kann als Indikator für eine funktionierende Unternehmenskommunikation dienen.

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Eine erfolgreiche SOA-Einführung beeinflusst die Unternehmensstruktur

- ⇒ Unterstützung durch den Vorstand/Unternehmensleitung entscheidend
- ▶ SOA Vorteile ergeben sich nicht sofort
 - ▶ steht im Kontrast zum Erfolgsdruck börsennotierter Unternehmen (schneller Erfolg erwartet)
 - ▶ vorsichtige, mittel- und langfristige Einführungsstrategie mit
 - ▶ ausreichend Zeit und Ressourcen

Organisatorische Aspekte und SOA im Unternehmenskontext

Organisatorische Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Wie werden die Kosten für eine Einführung von SOA verteilt?

- ▶ initiale Einführungskosten für Architektur und Infrastruktur
⇒ SOA-Governance
- ▶ Kosten für die Entwicklung einzelner neuer Services (zahlen sich erst mittelfristig aus):
mögliche Finanzierungsmodelle:
 - ▶ Der erste Nutzer, der einen neuen Service benötigt, zahlt die Entwicklung
 - ▶ Alle (absehbaren) Nutzer eines neuen Services teilen sich die Kosten der Entwicklung
 - ▶ Der Service-Anbieter trägt die Entwicklungskosten und berechnet die jeweilige Benutzung des Services (z.B. per Aufruf)
 - ▶ Die Entwicklungskosten werden aus einem allgemeinen Budget beglichen.

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen

Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

In der Praxis ist SOA nur ein Teil der gesamten Systemlandschaft

Wie passt sich SOA dort ein?

- ▶ Welche Rollenverteilung gibt es in SOA-basierten Systemen?
 - ▶ Wie lassen sich SOA-Landschaften darstellen?
- ▶ Wie werden die Aufgaben zwischen Frontend, Services und Backends verteilt?

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

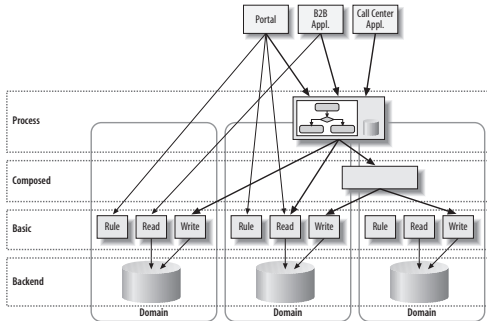
Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung



Unterschiedliche *Domänen* (Domains)

- z.B. Firma, Geschäftszweig, Geschäftseinheit, Abteilung, Team
- natürliche Einheit mit spezifischer Rolle und spezifischer **Verantwortlichkeit**

Backend:

- technisches System zusammen mit den Daten und Funktionalitäten der Domäne (konkret: Einzelsystem, Gruppe oder Teil eines Gesamtsystems)

Graphik: SOA in Practive, N. Josuttis

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

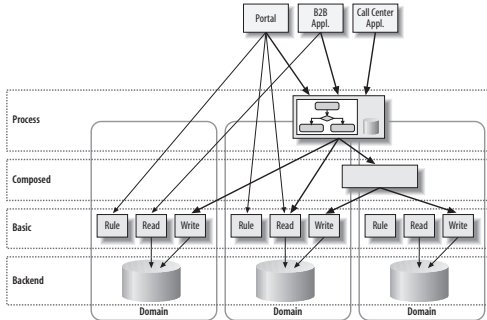
Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Wie sind die Verantwortlichkeiten verteilt?



► Basis-Services

liegen in der Verantwortlichkeit der Backend/Domänen-Abteilungen

► Zusammengesetzte Services und Prozess-Services

Querschnitts- oder übergreifende Abteilungen

- bestehende Abteilungen (*Auftragsverwaltung*)
- neue Abteilungen (*Prozess-Abteilung*) für alle übergreifende Prozesse zuständig

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

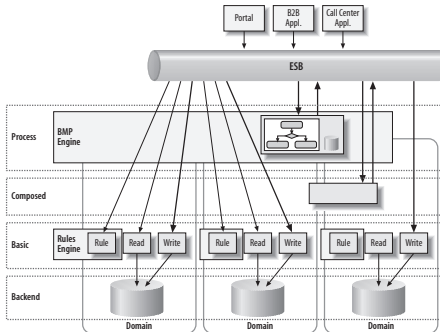
Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung



Hier dargestellte technische Aspekte:

- Kommunikation über den ESB
Keine Darstellung mehr, wie Basis-Services verwendet werden.
- Prozess-Services durch GPM-Tool verwaltet und auf einer Engine ausgeführt
- Implementierung der Basis-Logik-Services durch jeweils einen Regel-Prozessor (Rules-Engine)

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

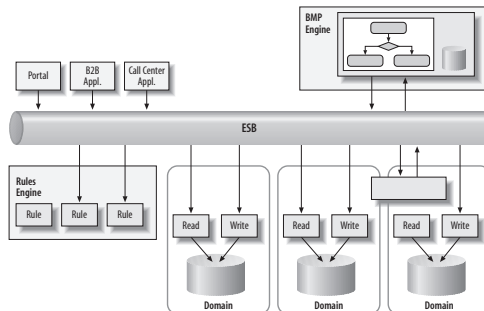
Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

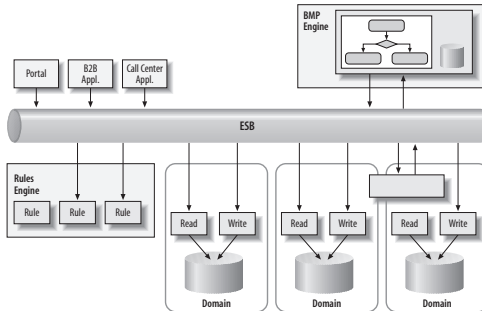
Zusammenfassung

Darstellung nach **technischen** Aspekten



- ▶ ESB steht im Zentrum der Betrachtung
- ▶ Domänen stellen nur Basis-Daten-Services (und zusammengesetzte Services) bereit
- ▶ Basis-Logik-Services und Prozess-Services befinden sich in getrennten Tools

Darstellung nach **technischen** Aspekten



Wie sind die Zuständigkeiten verteilt?

- Fachlich: Alle Daten- und Logik-Services zusammen in Domänen verwalten
- Technisch: alle Basis-Logik-Services zusammen in einer Regel-Engine verwalten
alle Prozess-Services an einer zentralen Stelle verwalten

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Wo liegt die Verantwortung für Prozesse und Geschäftsregeln?

- ▶ ganz in der Domäne
 - ▶ lesende, schreibende Services
 - ▶ und Basis–Logik-Services mit Geschäftsregeln
 - ▶ und Prozess–Services, die sich einer Domäne zuordnen lassen
- ▶ oder ganz in einem zentralen Geschäftsprozess–Management–Tool
 - ▶ Eine gemeinsame Sicht auf *alle* Prozesse
 - ▶ inhaltliche Abstimmung von Geschäftsregeln

Beides wird benötigt. In der Praxis wird es immer Mischformen geben (Heterogene Verteilte Systeme).

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

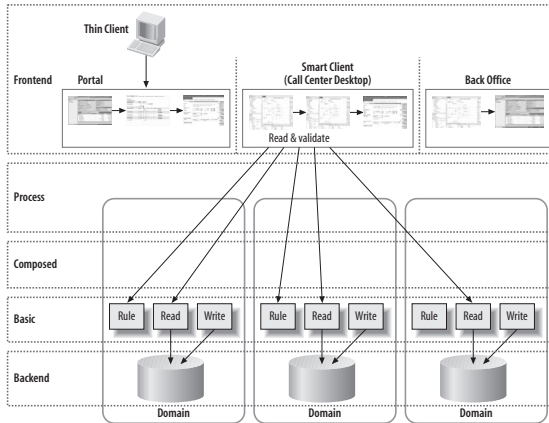
Zusammenfassung

- ▶ **Der Nutzer eines Systems ist aktiv.**
 - ▶ Er löst über das Frontend Aktionen in den Services aus.
 - ▶ Das Frontend bestimmt, wann welcher Service aufgerufen wird.
- ▶ **Services sind passiv** (auch Prozess–Services)
 - ▶ sie übernehmen keine Kontrolle
 - ▶ sie stellen passive Dienstleistungen zur Verfügung
 - ▶ sie können Arbeit delegieren
 - ▶ sie können nur Daten liefern (keine weiteren Aktivitäten)
 - ▶ sie können aktiv keine Interaktion mit Benutzern führen
 - ▶ Service-Aufruf ist wie ein Batch–Programm: Services erfüllen beim Aufruf eine Aufgabe
 - ▶ BPEL hat Charakter einer Batch–Sprache (siehe aber BPEL4People für Interaktionen mit BPEL)

Spezielles Paradigma für Frontends und Prozess–Services

- ▶ **Lesen:** Service, die Daten liefern werden aufgerufen, Daten dann angezeigt (oder Fehler behandelt)
- ▶ Sammeln der Daten erfolgt im Frontend (evtl. mit Vor–Validierung), außerhalb des schreibenden Services, durch Aufruf von Services die zulässige Werte liefern.
- ▶ **Schreiben:** Erfolgt mehrstufig....

1) Frontend liest Daten zur Beauskunftung und Vor-Validierung



Kein Service-Aufruf für das Validieren jedes einzelnen Eingabefeldes!

Graphik: *SOA in Praxise*, N. Josuttis

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

**Organisatorische
Aspekte**

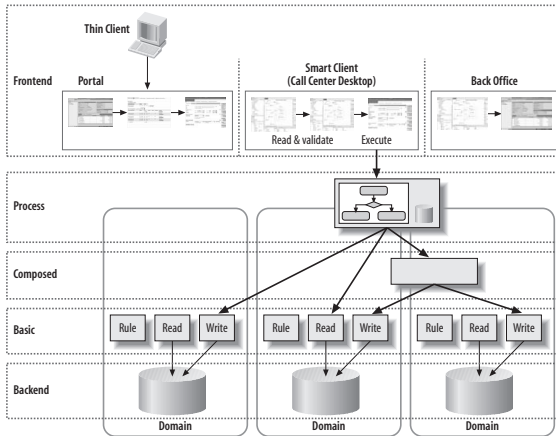
Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

2) Änderungen werden durch Aufruf der Prozess-Services für Backend-Updates als Ganzes geschrieben



Der laufende Prozess kann nicht mit dem Endanwender kommunizieren (kein Zugriff auf Frontend) sondern nur Resultat oder Fehler liefern oder in einen Fehlerzustand gehen.

Graphik: *SOA in Practive*, N. Josuttis

Organisatorische Aspekte und SOA im Unternehmenskontext

Organisatorische Aspekte

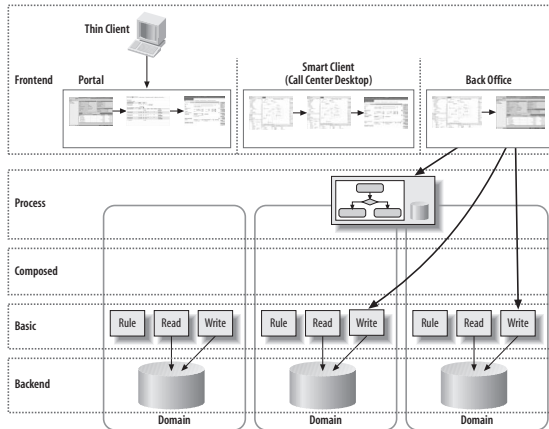
Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Fehlerbehandlung bei asynchronem Service-Aufruf



- ▶ laufender Prozess wird in einen Fehlerzustand versetzt.
- ▶ Back-Office setzt nach manueller Korrektur den unterbrochenen Prozess fort.

Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Wo findet die Konsistenzprüfung (*Vor-Validierung*) eingegebener Daten statt?

Validierung $\leftrightarrow$ Vor-Validierung

1. Validierung nur in den Backends

- ▶ Keine Prüfung in den Frontends.
- Fehlerhafte Daten werden erst sehr spät erkannt.
- Zu Frustration führende Benutzerführung

2. Validierung nur in den Frontends

- ▶ Backends verlassen sich darauf, nur richtige, konsistente Daten zu bekommen
- hohes Risiko, da Fehler oder unvollständige Prüfungen im Frontend zu Fehlern im Backend führen
- möglicherweise sind gar nicht alle Frontends bekannt

3. Redundante Validierung in Frontends und Backends

4. Zentrale Entscheidungskpunkte

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Validierung auf verschiedenen Ebenen

Wo findet die Konsistenzprüfung (*Vor-Validierung*) eingegebener Daten statt?

Validierung $\leftrightarrow$ Vor-Validierung

1. Validierung nur in den Backends

2. Validierung nur in den Frontends

3. Redundante Validierung in Frontends und Backends

- + Keine Fehler durch inkonsistente oder fehlerhafte Daten in den Backends
- + Gute Benutzerführung durch frühe Rückmeldung in den Frontends
- Doppelter Aufwand notwendig
- evtl. widersprüchliche Prüfungen in Front- und Backend

4. Zentrale Entscheidungskpunkte

- ▶ Validierung durch zentrale Services (decision points)
- ▶ Front- und Backends (enforcement points) bedienen sich dieser Services
- Gefahr zu feingranularer Prüfung mit zu vielen Service-Aufrufen

Wie kann man mehrere Frontend-Kanäle unterstützen.

- ▶ Übergabe eines Prozesses von einem Frontend zum nächsten
z. B. erst Online-Portal dann Call-Center
- ▶ Zusammenführen der einzelnen Prozess-Steuerungen der Frontends in einem zentralen System
 - ▶ Vereinheitlichung der Frontend-Prozesse
- ▶ Suchfunktion der Prozess-Engines, um Prozess nach unterschiedlichen Kriterien wiederzufinden
z. B. Wiederaufnahme anhand Auftragsnummer oder Kundenname
 - ▶ Problem: Umgang mit unterschiedlichen Suchkriterien
 - ▶ Wer ordnet dem Prozess die richtigen Kriterien zu?
 - ▶ Aufgabe der Prozess-Engine

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

**Organisatorische
Aspekte**

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung

Frontends mit mehreren Applikationen

Szenario: Bearbeiten der gleichen Daten mit unterschiedlichen Frontends auf dem selben Arbeitsplatz (z. B. Call-Center-Client)

- ▶ Typisch für getrennte Frontend–Entwicklung ohne Integration
- ▶ SOA: Jedes Frontend spricht Services selbst an.
- ▶ Wunsch: lokaler Austausch der Daten um Zeit und Ressource zu sparen

Optionen:

- ▶ direkter Datenaustausch
 - ▶ Querkommunikation zwischen den Frontends
 - ▶ Ein Frontend fragt, das andere antwortet
 - ▶ keine SOA–Struktur
- ▶ indirekter Datenaustausch
 - ▶ Daten, die ein Frontend anfordert, stehen auch anderen zur Verfügung
 - ▶ lokaler Cache für Service–Aufrufe auf Nutzerseite
 - ▶ im ESB implementierbar
 - ▶ minimale lokale Kommunikation nötig
- ▶ Als Optimierung anzusehen,
die erst erfolgen sollte, wenn es Probleme gibt.

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle

Front- und Backends

Zusammenfassung

Organisatorische Aspekte

Rollen und Organisationen
Finanzierungsmodelle

SOA im Unternehmenskontext

Architekturmodelle
Auswirkungen auf Front- und Backends

- ▶ Fragen?
- ▶ nächste Woche: Message–Exchange Patterns

**Organisatorische
Aspekte und
SOA im Unter-
nehmenskontext**

Organisatorische
Aspekte

Rollen und Organisationen
Finanzierungsmodelle

im Unternehmen

Architekturmodelle
Front- und Backends

Zusammenfassung