

# Service-orientierte Software-Architekturen

Prof. Dr. U. Hoffmann  
FH Wedel

Services  
und  
Lose Kopplung

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung

## Einführung in Service-orientierte Architekturen

- ▶ technische Konzepte
- ▶ betriebliche Konzepte

### Weitere Quellen:

- ▶ Einführender SOA-Artikel *Serviceorientierte Architektur* von Richter, Haller, Schrey erschienen im Informatik-Spektrum (Oktober 2005) online verfügbar bei der Gesellschaft für Informatik im *Informatiklexikon* unter [http://www.gi-ev.de/no\\_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/serviceorientierte-architektur-118.html](http://www.gi-ev.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/serviceorientierte-architektur-118.html)
- ▶ Detaillierte SOA-Literatur findet sich online auch in den IBM-Redbooks unter <http://www.redbooks.ibm.com/>
- ▶ Software-Engineering-Radio <http://www.se-radio.net/tags/soa> Episoden über SOA

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung

## Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

## Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

### Services und Lose Kopplung

#### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

#### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

#### Zusammenfassung

# Was ist ein Service?

Wie für den Begriff *Service-orientierte Architektur* selbst, gibt es auch keine einheitliche Definition dafür, was ein *Service* ist.

Der Begriff *Service* wird in Wörterbüchern als *Die Durchführung einer Aufgabe (einer Tätigkeit) von jemandem für einen anderen* bezeichnet. OASIS–SOA–Referenzmodell

Im Kontext von Unternehmensarchitekturen (also Systemen oder organisatorischen Architekturen oder Modellierungen) bezieht sich der Begriff *Service* auf eine diskrete Menge von fortlaufenden und autonomen fachlichen oder technischen Funktionalitäten...  
wikipedia.com, 2007-07-06

## Services und Lose Kopplung

### Services

#### Definition

Schnittstellen

Zusätzliche Eigenschaften

#### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

#### Zusammenfassung

„Services sind eine IT–Repräsentation von fachlicher Funktionalität, die durch eine (wohldefinierte) Schnittstelle beschrieben wird. Services sollten außerdem in sich abgeschlossen sein (autark sein und für sich selbst stehen).

Nicolai Josuttis

## Services

- ▶ repräsentieren einen *natürlichen* Ausschnitt der **fachlichen Funktionalität**.
- ▶ sind in sich abgeschlossen (*self-contained*).
- ▶ entsprechen einer tatsächlichen Funktionalität im Alltag.
- ▶ sollen für Fachleute ohne technisches Detailwissen verständlich sein.
- ▶ werden in der *Geschäftsprozessmodellierung (business process modeling)* identifiziert und entworfen.

## Services und Lose Kopplung

### Services

#### Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

„Im Grunde ist SOA eine Softwarearchitektur, die mit einer Schnittstellen–Definition beginnt und die gesamte Anwendungsstruktur auf eine Struktur von Schnittstellen, Schnittstellen–Implementierungen und Schnittstellen–Aufrufen aufbaut. SOA würde eigentlich besser *Schnittstellenorientierte Architektur* heißen.“

V.Natis (*Service–Oriented Architecture Scenario*, Gartner, 2003)

Anwendung des Prinzips der *Modularisierung* des klassischen Software–Engineering im großen Stil.

Beschreibungen von Schnittstellen bestehen aus zwei Teilen:

1. **syntaktische Schnittstellenbeschreibung**

legt die Signatur fest, also die Parameter– und Ergebnistypen der Operationen und gibt mglw. auftretende Exceptions an

2. **semantische Schnittstellenbeschreibung**

legte das (relevante) Verhalten des Moduls (Services) fest

Eine Schnittstelle für die beide Teile angemessen vorliegen, nennt man *wohldefiniert*.

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

## Beispiel

```
bearbeiteKunde(  
  aktion,      // anlegen, lesen, aendern, loeschen  
  id,          // Kundennummer, oder null  
  name,        // neuer Kundename oder null  
  adresse,     // neues Kundenadresse oder null  
  konto)       // neue Bankverbindung oder null
```

- ▶ Fachliche Schnittstelle, die die Bearbeitung von Kunden erlaubt
- ▶ Diese Schnittstelle ist **technisch getrieben**.
- ▶ Sie kehrt technische Details (Umsetzung in Datenbankzugriffe) nach außen.
  
- ▶ Wie sieht eine **fachlich getriebene** Schnittstelle aus?

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

## Beispiel

```
erzeugeKunde (
  name,           // neuer Kundenname
  adresse ,      // neue Kundenadresse
  konto)

leseKunde ( id ) // Kundennummer

aendereKunde (
  id ,           // Kundennummer
  konto)        // neue Bankverbindung

aendereAdresse (
  id ,           // Kundennummer
  adresse ,     // neue Kundenadresse
  pruefen ,    // true: Adresse pruefen
  korrigieren  // true: Adresse ggf. korrigieren
)

loescheKunde ( id ) // Kundennummer
```

Unterschiede zwischen den Schnittstellen:

1. eine generische Operation vs. fünf spezielle Operationen
2. unterschiedliche Granularität der Operationen
3. zusätzliche Parameter bei einigen Operationen

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

**Design by Contract** ist ein Software–Entwicklungs– Konzept von Bertrand Meyer realisiert in der Programmiersprache Eiffel:

- ▶ für Klassen werden **Klasseninvarianten** definiert: Eigenschaften, die für Objekte der Klasse immer gelten (wenn nicht gerade eine Methode ausgeführt wird).
- ▶ für jede Methode wird definiert:
  - ▶ **Vorbedingung** (pre condition)  
Bedingung, die der Nutzer bei Aufruf der Methode sicher stellen muss
  - ▶ **Nachbedingung** (post condition)  
Garantie der Anbieters über Eigenschaften, die gelten wenn die Operation des Services erfolgreich abgeschlossen wurde.

Übertragung auf Services zur Beschreibung der semantischen Eigenschaften der Schnittstelle.

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

Die Beschreibung eines Services muss neben seiner Schnittstelle auch die **Nichtfunktionale Eigenschaften** umfassen:

- ▶ zugesicherte Laufzeit
- ▶ Verfügbarkeit
- ▶ Zuverlässigkeit
- ▶ ...

Hierzu wird die

- ▶ die sog. **Service-Qualität, (Quality of Service, QoS)**  
(etwa: *Wann steht der Service mit welcher Leistungsfähigkeit zur Verfügung?*) definiert und
- ▶ in sog. **Service-Level-Agreements (SLA)**  
(Vereinbarungen über die Qualität der Services) festgelegt.

Verbindliche Beschreibung eines Services durch  
**Verträge** (*contracts*):

- ▶ geschlossen zwischen
  - ▶ einem spezifischen Service–Anbieter und
  - ▶ einem spezifischen Service–Nutzer
- ▶ enthalten *Alles was man wissen muss, wenn man den Service benutzen will.*
- ▶ lassen keine Fragen offen
- ▶ umfassen die vollständige Spezifikation des Services
  - ▶ syntaktische und semantische Schnittstellenbeschreibung
  - ▶ Beschreibung der nichtfunktionalen Eigenschaften

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

Services sollen *abgeschlossen* sein  
(*unabhängig, autonom, self-contained*)

- ▶ Abhängigkeiten sind immer vorhanden
  - ▶ z.B. durch die Verwendung gemeinsamer Datentypen
- ▶ Operationen unabhängig
  - ▶ von anderen Services
  - ▶ vom Systemzuständen
- ▶ Ziel: Minimierung von Abhängigkeiten

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

In der SOA–Gemeinde werden weitere Fragestellungen über Services diskutiert:

## Müssen Services

- ▶ als Web–Services implementiert werden?
- ▶ zustandslos (stateless) sein?
- ▶ grobgranular sein?
- ▶ komponier/aggregierbar sein?
- ▶ ...

Sind diese geforderten Eigenschaften für Services zwingend?

Die Eigenschaften, die Services haben müssen, hängen stark von der jeweilige Situation ab.

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

### Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

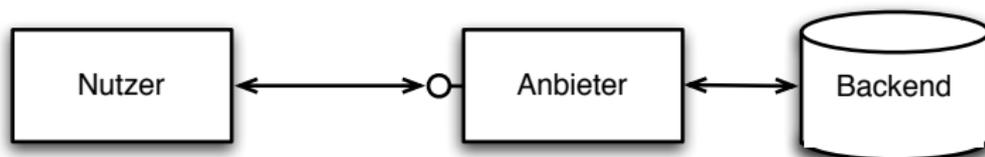
### Zusammenfassung

## Sollen Services zustandslos sein?

Was heißt hier eigentlich zustandslos?

Ein Zustand ist ja immer im Spiel.

- ▶ Wo existiert der Zustand?
- ▶ Wie lange existiert der Zustand?
- ▶ Ist es ein fachlicher oder ein technischer Zustand?



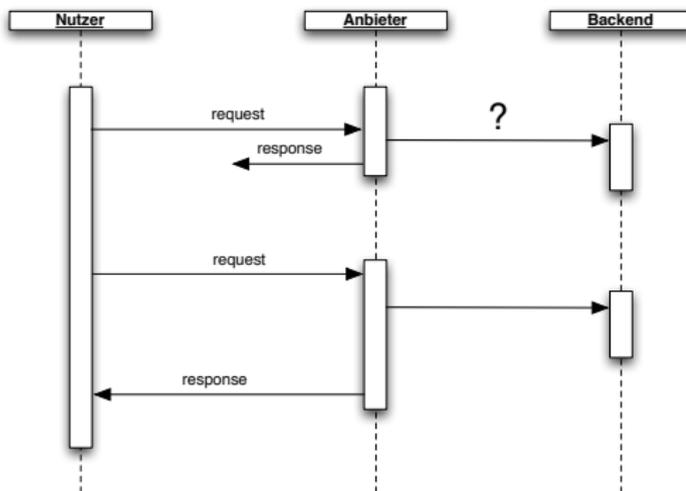
## Verhalten in Fehlersituationen in verteilten Systemen

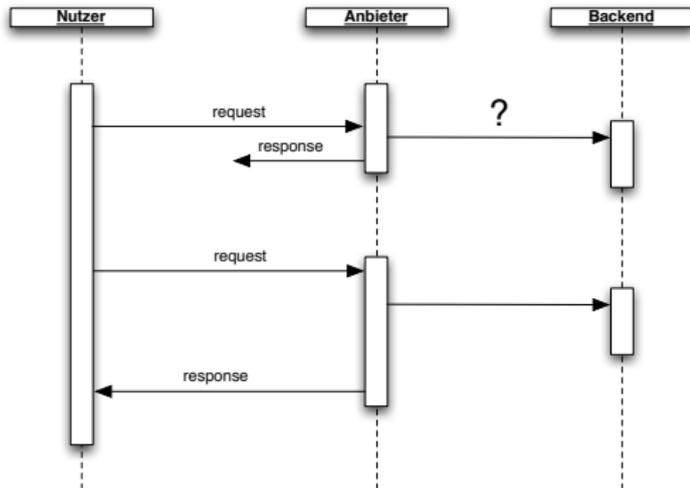
Änderungen ohne Bestätigung. In welchem Zustand ist dann das System?

1. Anbieter hat Änderung geschrieben, Bestätigung ging verloren
2. Anbieter hat keine Änderung geschrieben.

Erneutes Schreiben (dieses mal mit Bestätigung)

1. Änderung wird ein zweites mal durchgeführt
2. Änderung wird erstmalig vorgenommen





## Definition (Idempotenz)

Ein Service, bei dem mehrfache (erfolgreiche) Zugriffe, die gleiche Wirkung wie ein einmaliger (erfolgreicher) Zugriff haben, nennt man **idempotent**.

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

Wie fein– oder grobgranular soll ein Service sein?

Wie fein oder grob soll auf Daten zugegriffen werden können?

- ▶ Abstraktion kostet Laufzeit
- ⇒ Alle relevanten Daten auf einmal übertragen
- ▶ grobe Granularität hilft für Trennung interner Datenstrukturen und externer Schnittstelle
- ▶ Gefahr zu viel zu übertragen
- ▶ Granularität keine wirklich messbare Größe
  
- ▶ gesunde Abwägung in der konkreten Situation

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

## Services

- ▶ sind nicht nur elementar
  - ▶ können andere Services aufrufen
  - ▶ mehrere Services werden zu höherwertigen Services zusammengesetzt
  - ▶ nicht alle Services haben gleiche Eigenschaften
- ⇒ Klassifizierung von Services
- ▶ Geschäftsprozessmodellierung

### Services und Lose Kopplung

#### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

#### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

#### Zusammenfassung

Wie wird die Existenz eines Services in Erfahrung gebracht?

- ▶ Bekanntgabe an zentraler oder öffentlicher Stelle (Repository/Registry)
- ▶ Absprachen zwischen Anbieter und Nutzer
- ▶ Hörensagen, Empfehlungen

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

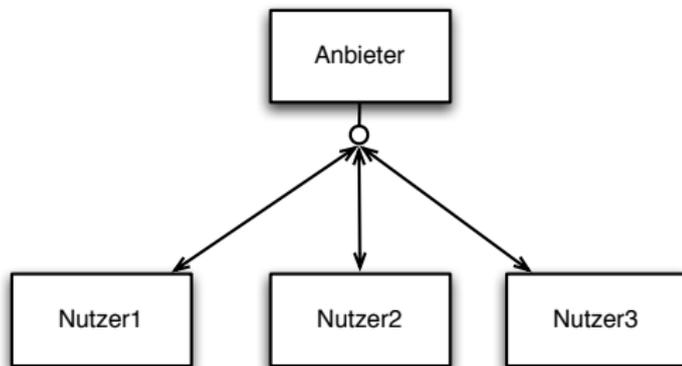
Kompensation

Infrastruktur

### Zusammenfassung

Allgemeines Ziel der Software-Entwicklung:

- ▶ Doppelt-Entwicklungen vermeiden
- ▶ Weniger Aufwand, höhere Konsistenz



Wiederverwendbarkeit wird als Vorteil von SOA gepriesen

- ▶ In der Praxis nicht immer erreicht
- ▶ Erhöht Abhängigkeit
- ▶ Problem: Performance

## Technische Services

- ▶ repräsentieren *keine* abgeschlossene fachliche Funktionalität
- ▶ dienen technischen Aspekten der Infrastruktur
  - ▶ Verfügbarkeitsprüfung für Services
  - ▶ Konfiguration der Systemlandschaft
  - ▶ Steuerung der Systemlandschaft
  - ▶ Protokollierungsaufgaben
- ▶ spezielle Abstraktionsebene
- ▶ von *normalen* Services deutlich zu unterscheiden

Ein Service ist die IT–Repräsentation einer in sich abgeschlossenen fachlichen Funktionalität.

Nicolai Josuttis

- ▶ technische Details verbergen
  - ▶ für fachliche Experten verständlich
  - ▶ technisch bietet ein Service eine Schnittstelle über die Nutzer und Anbieter Nachrichten austauschen
  - ▶ vollständige Beschreibung: *Vertrag*:
    - ▶ *wohldefinierte Schnittstelle* (Syntax und Semantik)
    - ▶ geschlossen zwischen konkretem Nutzer und konkretem Anbieter
    - ▶ umfasst auch nichtfunktionale Eigenschaften (QoS, SLAs)
  - ▶ weitere Eigenschaften sind situationsbedingt (nicht essentiell)
- ⇒ Klassifizierung von Services

## Services und Lose Kopplung

### Services

Definition

Schnittstellen

Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

Kommunikation

Datentypen

Kompensation

Infrastruktur

### Zusammenfassung

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung

- ▶ SOA: große, verteilte Systeme
- ▶ Systeme müssen erweiterbar sein.
- ▶ Systeme müssen tolerant gegenüber Fehlern sein.
- ▶ Kerngeschäftsprozesse müssen funktionieren.
- ▶ Probleme dürfen nur geringe Auswirkung haben.
- ▶ Systeme unterliegen ständigen Änderungen.

## ⇒ Lose Kopplung

Ziel: Minimieren von Abhängigkeiten

Welche Formen von Loser Kopplung lassen sich unterscheiden?

	enge Kopplung	lose Kopplung
Physikalische Verbindung	Punkt-zu-Punkt	über Vermittler
Kommunikationsstil	synchron	asynchron
Datenmodell	komplexe gemeinsame Typen	nur einfache gemeinsame Typen
Typsystem	streng	schwach
Binding	statisch	dynamisch
Plattformspezifika	stark/viel	schwach/wenig
Interaktionsmuster	über komplexe Objekt-Bäume navigieren	datenzentrierte autonome Nachrichten
Transaktionssicherheit	2-Phase-Commit	Kompensation
Kontrolle fachlicher Logik	zentrale Kontrolle	verteilte Kontrolle
Deployment	gleichzeitig	zu verschiedenen Zeitpunkten
Versionierung	explizite Upgrades	implizite Upgrades

Abwägung zwischen

- ▶ gewünschter Loser Kopplung und
- ▶ dem erhöhten Entwicklungs- und Pflege-Aufwand

Quelle: *SOA in Practice*, N. Josuttis, *Enterprise SOA*, Krafzig et. al.

Services  
und  
Lose Kopplung

Services

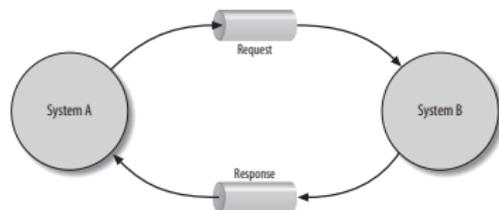
Definition  
Schnittstellen  
Zusätzliche Eigenschaften

Lose Kopplung

unterschiedliche Formen

Kommunikation  
Datentypen  
Kompensation  
Infrastruktur

Zusammenfassung



## Entkopplung durch asynchrone Kommunikation

- ▶ System A stellt eine Anfrage (request).
- ▶ System B muss zu diesem Zeitpunkt nicht verfügbar sein.
- ▶ System A muss nicht auf die Antwort von System B warten, um weiterzuarbeiten.
- ▶ System B bearbeitet die Anfrage, sobald es verfügbar ist.
- ▶ System B schickt Antwort (response) sobald die Anfrage abschließend bearbeitet ist.
- ▶ System A muss zu diesem Zeitpunkt nicht verfügbar sein.
- ▶ System A verarbeitet diese Antwort, wenn es passend ist.

Unterschiedliche Auswirkungen auf Nutzer, Infrastruktur und Anbieter

**Lose Kopplung erhöht die System-Komplexität.**

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

#### Kommunikation

- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung

Asynchrone Kommunikation erhöht die System-Komplexität:

- ▶ Zeitpunkt des Eintreffens der Anfrage ist unbekannt
- ▶ Behandlung von langen Zustellzeiten und Fehlzustellungen
- ▶ Zuordnung der später eintreffenden Antwort zum ursprünglichen Vorgang
- ▶ Wiederherstellen des zugehörigen Kontextes
- ▶ Reihenfolgen bei vielen zusammengehörigen asynchronen Nachrichten
- ▶ Problem beim Testen und Debuggen (nur noch statistische Effekte, Gefahr von Race-Conditions)

**Vor- und Nachteile abwägen.**

## Services und Lose Kopplung

### Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

#### Kommunikation

Datentypen  
Kompensation  
Infrastruktur

### Zusammenfassung

Wie findet eine Service–Nutzer den zugehörigen Service–Anbieter?

Enge Kopplung:

- ▶ Punkt–zu–Punkt–Verbindungen
- ▶ Zieladresse ist dem Nutzer bekannt
- ▶ Bei *Umzug* des Anbieters müssen Nutzer angepasst werden.
- ▶ kein Lastverteilung möglich
- ▶ keine Ausfallsicherheit

Vermittler ermittelt den Kommunikationspartner:

- ▶ **im Vorfeld der Aufrufe** (Broker, Name–Server)
  - ▶ Nutzer macht dann Punkt–zu–Punkt Aufruf
  - ▶ zusätzlicher Aufwand beim Nutzers (Befragung des Brokers)
- ▶ **während des Aufrufs**
  - ▶ Senden an symbolischen Namen
  - ▶ Infrastruktur (Netzwerk, Middleware, ESB) sorgt für die Zustellung
  - ▶ Umstrukturierungen haben für Nutzer keine Auswirkungen

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen

#### Kommunikation

- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung

## Gemeinsame Datentypen

- ▶ können von vielen Systemen verwendet werden
- ▶ erhöhen die Wiederverwendbarkeit
- ▶ erleichtern die Programmierung und Pflege
- ▶ Vereinheitlichung der Datentypen über mehrere Systeme:  
*Harmonisierung*
- ▶ objekt-orientiertes Ziel: einheitliches, fachliches Objektmodell (business object model, BOM).

### Quelle von Fehlschlägen

- ▶ Einigung auf gemeinsame Datentypen ist ein extrem schwieriger Prozess (Gremienabstimmungen sind lähmend und langwierig.)
- ▶ Unterschiedliche Entwicklungsgeschwindigkeiten, Änderungen an gemeinsamen Datentypen betreffen alle, die sie verwenden.

## Harmonisierung von Datentypen führt zu enger Kopplung

### Services und Lose Kopplung

#### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

#### Lose Kopplung

- unterschiedliche Formen
- Kommunikation

#### Datentypen

- Kompensation
- Infrastruktur

#### Zusammenfassung

- ▶ Abhängigkeiten wachsen mit der Anzahl der beteiligten Systeme.
  - ▶ Gemeinsame fachliche Geschäftsobjektmodelle skalieren nicht, da die Kopplung zu groß wird.
- ⇒ Für große verteilte Systeme, Tatsache akzeptieren:
- ▶ Datentypen sind nicht harmonisiert.
  - ▶ Datentypen sind nicht harmonisierbar.



- ▶ Entkopplung: Datentypen unterscheiden sich.
  - ▶ Es muss eine Umsetzung (Mapping) stattfinden
  - ▶ positive Eigenschaft, da sie zur Entkopplung führt
- ▶ gängige Praxis:
  - ▶ Anbieter definiert Datentypen
  - ▶ Nutzer konvertieren schnell in eigene, interne Datentypen.

## gemeinsame *fundamentale* Datentypen

- ▶ grundlegende Basis für höhere fachliche Datentypen
- ▶ starke Abhängigkeit bei vielen/allen Systemen
- ▶ ändern sich selten, de facto keine Änderung möglich
- ▶ Beispiele für Basisdatentypen:
  - ▶ technische: String, Integer, Datum/Zeit
  - ▶ fachliche: Telefonnummer
  - ▶ Gegenbeispiel: Adresse

## Harmonisierung ist begrüßenswert

- ▶ Harmonisierung wo möglich, nicht dogmatisch
- ▶ SOA üblich:
  - ▶ Composed Service bietet Schnittstelle mit Datentypen, um etwa Änderung von Adressdaten unterschiedlicher Struktur in verschiedenen Backends vorzunehmen.
  - ▶ Vorschlag für einen gemeinsamen Datentyp, kein Zwang

### Services und Lose Kopplung

#### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

#### Lose Kopplung

- unterschiedliche Formen
- Kommunikation

#### Datentypen

- Kompensation
- Infrastruktur

#### Zusammenfassung

Welche Datenstrukturen werden in Schnittstellen definiert?

- ▶ Nur Strings in Schnittstellen verwenden
- ▶ andere elementare Datentypen (Integer, Float, Date/Time,...)
- ▶ Aufzählungstypen
- ▶ Einschränkung von Wertebereichen
- ▶ Definitionsmöglichkeit für Records
- ▶ Listen/Arrays von Werten
- ▶ Beziehungen zwischen Datentypen (Vererbung, Erweiterung, Polymorphie)

Komplexe Datenstrukturen

- ▶ erlauben Abstrakte Formulierung der Schnittstelle
- ▶ Schwierigkeiten der Abbildung auf Plattformen, die dies nicht unterstützen
- ▶ Einmal eingeführte komplexe Typen lassen sich nur schwer zurücknehmen

Empfehlung:

- ▶ elementare Datentypen + Records + Listen
- ▶ Aufzählungstypen, Vererbung, Polymorphie vermeiden

Services  
und  
Lose Kopplung

Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

Lose Kopplung

unterschiedliche Formen  
Kommunikation

Datentypen

Kompensation  
Infrastruktur

Zusammenfassung

Änderungen an Schnittstellen sollen nur die wirklich Beteiligten betreffen.

- ▶ konkrete Datentypen:
  - ▶ einfacher zu verstehen
  - ▶ Änderungen in den Datenstrukturen müssen die Schnittstelle erweitern
- ▶ generische Datentypen (z.B. key-value-Paare)
  - ▶ Erweiterung gut möglich, Schnittstelle kann unverändert bleiben
  - ▶ Benutzung eher beliebig, daher feheranfälliger

Empfehlung: Generische Erweiterungsmöglichkeit vorsehen

- ▶ Erweiterungen ermöglichen, auch wenn die Schnittstelle nicht mehr geändert werden darf.

## Vor- und Nachteile abwägen

### Services und Lose Kopplung

#### Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

#### Lose Kopplung

unterschiedliche Formen  
Kommunikation

#### Datentypen

Kompensation  
Infrastruktur

#### Zusammenfassung

## Services und Lose Kopplung

### Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen  
Kommunikation

### Datentypen

Kompensation  
Infrastruktur

### Zusammenfassung

## Was geschieht bei einer Schnittstellenänderung?

- ▶ Neue Version der Schnittstelle bereitstellen
- ▶ Abhängigkeit der Nutzer von der Schnittstelle
- ▶ Mechanismen, dass Nutzer bei *rückwärtskompatiblen* Änderungen nicht reagieren müssen
- ▶ Mehrere Versionen zur Verfügung stellen

*Transaktionssicherheit*: Wie kann bei einer verteilten Transaktion der gemeinsame Erfolg oder das gemeinsame Fehlschlagen beschlossen werden.

Klassischer Ansatz (enge Kopplung)

- ▶ Kommunikation mit Hilfe des *2-Phase-Commit*-Protokolls
    1. Änderungen vorbereiten, warten auf Freigabe
    2. letzte Freigabe kommunizieren, Änderungen übernehmen
  - ▶ alle beteiligten Systeme müssen 2PC unterstützen
  - ▶ alle beteiligten Systeme müssen online sein
  - ▶ Möglichkeit von Deadlocks
- ⇒ in der Praxis seltener eingesetzt als zu erwarten

## Services und Lose Kopplung

### Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

### Lose Kopplung

unterschiedliche Formen  
Kommunikation  
Datentypen  
Kompensation  
Infrastruktur

### Zusammenfassung

## Lose gekoppelte Alternative zu 2PC: *Kompensation*

- ▶ Änderungen erfolgen nacheinander
- ▶ Bei Fehlschlagen späterer Änderungen erfolgt geeignete Kompensation früherer Änderungen
  - ▶ Rücknahme (Rollback)
  - ▶ Korrektur (Storno)
  - ▶ Erneuter Versuch der späten Änderung
  - ▶ Manuelle Nacharbeitung
- ▶ explizite Programmierung der Umkehr-Mechanismen
- ▶ keine synchrone Änderung nötig

Unterstützung von Kompensation in der Sprache BPEL für Geschäftsprozessmodellierung

### Services und Lose Kopplung

#### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

#### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen

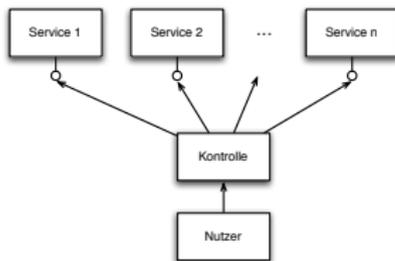
#### Kompensation

- Infrastruktur

#### Zusammenfassung

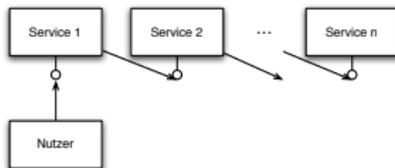
Wie erfolgt die Abwicklung (Start, Steuerung) eines Geschäftsprozesses?

► Durch eine zentrale Komponente



- Ausfall hat große Auswirkung
- Gefahr der Überlastung

► dezentrale Kontrolle



- Bearbeitungszustand schwer zu ermitteln

Services

Definition  
Schnittstellen  
Zusätzliche Eigenschaften

Lose Kopplung

unterschiedliche Formen  
Kommunikation  
Datentypen  
Kompensation

Infrastruktur

Zusammenfassung

## Wie werden Services in Betrieb genommen?

- ▶ Änderungen werden an vielen Systemen gleichzeitig ausgerollt
  - ▶ Synchronisationsaufwand
  - ⇒ hohe Abhängigkeit
  
- ▶ Änderungen erfolgen zu unterschiedlichen Zeiten
  - ▶ geringeren Abhängigkeit (andere Releasezyklen möglich)
  - ▶ Notwendigkeit der Migration

### Services und Lose Kopplung

#### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

#### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation

#### Infrastruktur

#### Zusammenfassung

## Lose Kopplung

- ▶ ein fundamentales SOA–Konzept  
Minimierung von Abhängigkeiten
  - ▶ Verschiedene Formen  
den Umständen angemessene Mischung aus loser und enger  
Kopplung finden
  - ▶ bringt je nach Form unterschiedliche Nachteile mit sich.
- ⇒ darf nicht Selbstzweck werden
- ▶ Mapping von Daten ist in großen Systemen eine *gute*  
Eigenschaft

## Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

## Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

- ▶ Fragen?
- ▶ nächste Woche:  
Enterprise Service Bus und Service-Klassifizierung

## Services und Lose Kopplung

### Services

- Definition
- Schnittstellen
- Zusätzliche Eigenschaften

### Lose Kopplung

- unterschiedliche Formen
- Kommunikation
- Datentypen
- Kompensation
- Infrastruktur

### Zusammenfassung