TKeasy

Schichtenarchitekturen und ihre Auswirkungen auf die objektorientierte Modellierung

Software Engineering 2007, 27.-30.3.2007, Hamburg

Ralf Degner, Frank Griffel

Techniker Krankenkasse
Gesund in die Zukunft.

Überblick

- Das Projekt TKeasy
- Technische Schichten und ihr Probleme
- Logische Schichten
- Trennung der Verantworlichkeiten – Schichten als Abstraktionen
- Objektorientierung ermöglichen

Das Projekt - TKeasy

- Java basiertes unternehmensweites EDV-System
- Einheitliches Framework für alle Anwendungen des Unternehmens
- Eigenentwicklung von Application-Server und Persistenz-Dienst
- Auf SWING basierende GUI-Framework
- Einbindung der Altsysteme über Objektmapping und Services
- Ziel: durchgängiger Einsatz von OO-Methoden und Technologien

TKeasy in Zahlen (ca. pro Tag):

- Über 7000 parallele Clients
- 1 Millionen Anwendungen
- 150 Millionen erzeugte Geschäftsobjekte
- 2 Milliarden Methodenzugriffe
- 15 Millionen Großrechnertransaktionen (DB-System)

Technische Schichten (I)

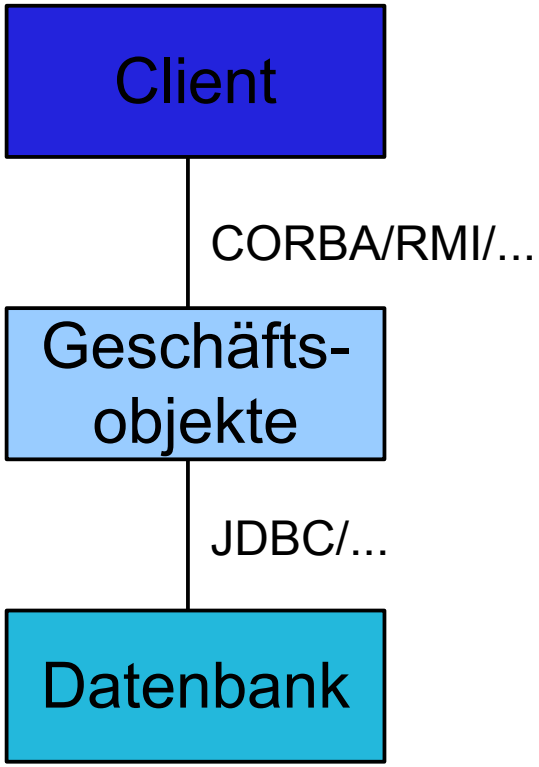
Ausrichtung von Schichten an technischen Grenzen => jede Schicht auf eigenem System

Beispiel: Geschäftsobjekt *Kunde*

Methoden des *Kunden*:

- getName()
- getAdresse()
- getGeburtsdatum()
- ...

Darstellung des *Kunden* erfordert viele Methodenzugriffe auf den *Kunden*.



Technische Schichten (II)

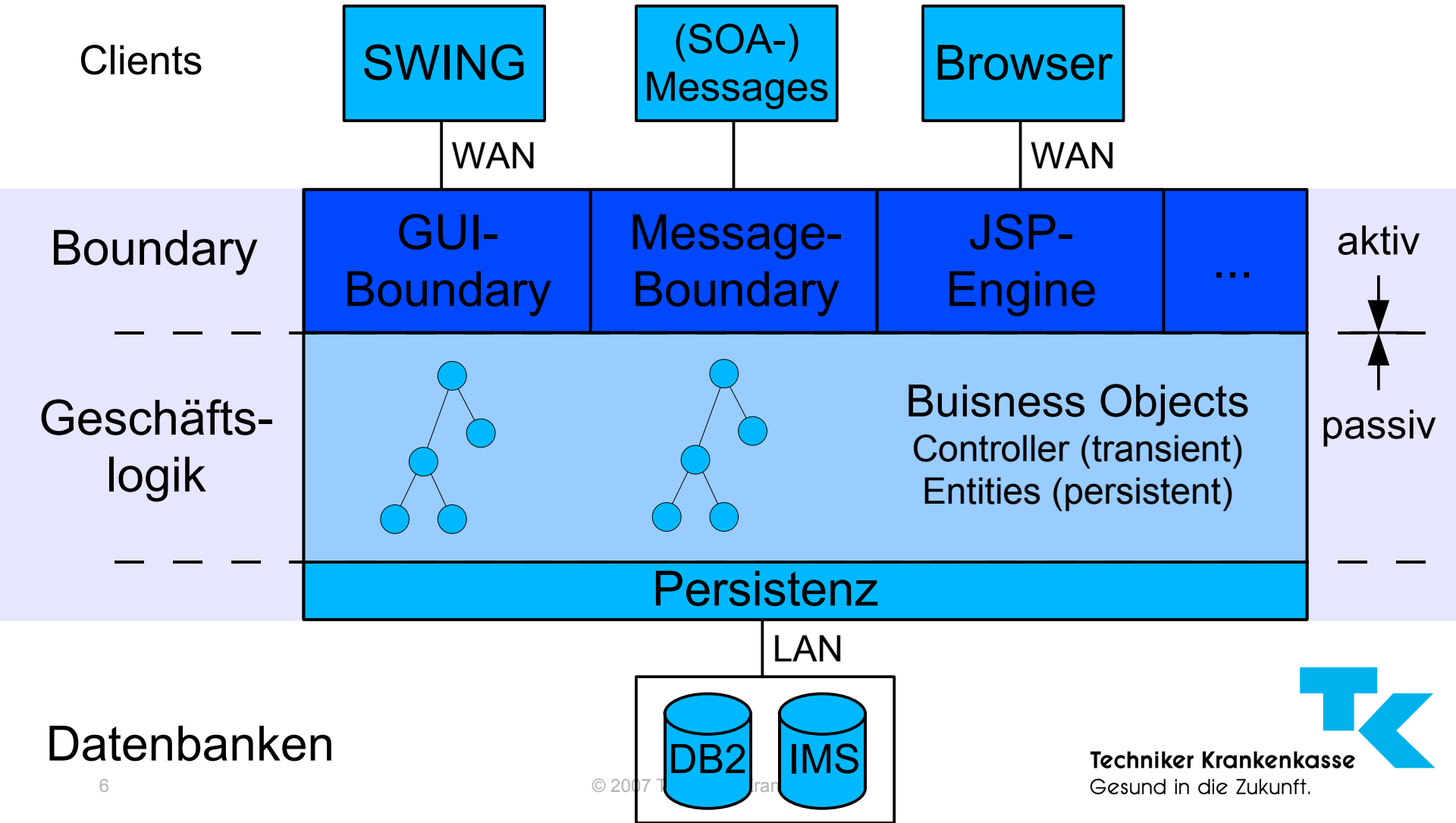
Schlechte Performance durch viele „feine“ Zugriffe auf Geschäftsobjekte (BOs)

Lösungsansatz: weniger Zugriffe durch andere Schneidung der Methoden, z.B. getStammdaten()

Folgen:

- Methode an Bedürfnissen eines Nutzers ausgerichtet
- Unterschiedliche Klienten führen zu unterschiedlichen Methoden
- Objekt konzentriert sich damit nicht auf „sein“ Geschäft
- Keine klare Trennung der Verantwortlichkeiten der Schichten
- Feingranulare Modellierung eingeschränkt

Logische Schichten
Boundary – BOs - Persistenz



Logische Schichten

- Client hat Adapter (Boundary) in der VM der Geschäftsobjekte
- Feingranulare Zugriffe von Boundary auf BOs ohne Einschränkung
- Boundary nutzt BOs und beliefert Client nach dessen konkreten Bedürfnissen
- Boundary gehört logisch zur Schicht Präsentation

=> Bedürfnisse des Clients optimal unterstützt ohne dadurch BO-Design zu beeinflussen

=> Klare Trennung der Verantwortlichkeiten zwischen Geschäftslogik und Darstellung

Trennung der Verantwortlichkeiten – Schichten als Abstraktionen

- Jede Schicht stellt eine Abstraktion von Aufgaben und Verantwortlichkeiten dar.
- „Jede kümmert sich um ihren Kram (und hält sich aus dem der anderen raus)“
- Schichten müssen hierfür „undurchlässig“ sein

Fast jedes Projekt plant dies, kaum eines hält es durch.
(automatisierte Prüfungen hilfreich)

Objektorientierung ermöglichen

- Schneidung der Schichten hat starken Einfluss auf die Freiheit OO einzusetzen
- Auch Frameworks für Security, Transaktionshandling und Persistenz beeinflussen OO-Modellierung stark (z.B. Einschränkung durch Laufzeit-Overhead)

Wichtige Eigenschaften für vollwertiges OO in großen Systemen:

- Feingranulare Modellierung
- Vererbung
- Exception-Handling
- Generische Referenzen
- Geschachtelte Transaktionen auf BOs
- ...

Zusammenfassung

- Wahl der Schichten hat starke Rückwirkung auf die OO-Modellierung
- Grenzen von logischen Schichten unabhängig von Systemgrenzen
- Schichten zur klaren Trennung von Verantwortlichkeiten

Vielen herzlichen Dank

für Ihre Aufmerksamkeit

Frank.Griffel@tk-online.de

Ralf.Degner@tk-online.de

Techniker Krankenkasse

Gesund in die Zukunft.