Die AspectIX Middleware-Plattform

Franz J. Hauck – Verteilte Systeme, Univ. Ulm

1 Arbeitsbereiche

1.1 QoS-Team

- Dienstgüte bei der Übertragung multimedialer Daten für mobile Teilnehmer
  - nahtlose Reservierung mit RSVP bei Mobile IP
  - optimierte Dienstgüteverhandlung von End zu End bei mobilen Teilnehmern
  - adaptives Transcoding und Filtern multimedialer Datenströme in mobilen Netzen

  - drei wissenschaftliche Mitarbeiter
  - DFG- und EU-Förderung

1.2 AspectIX-Team

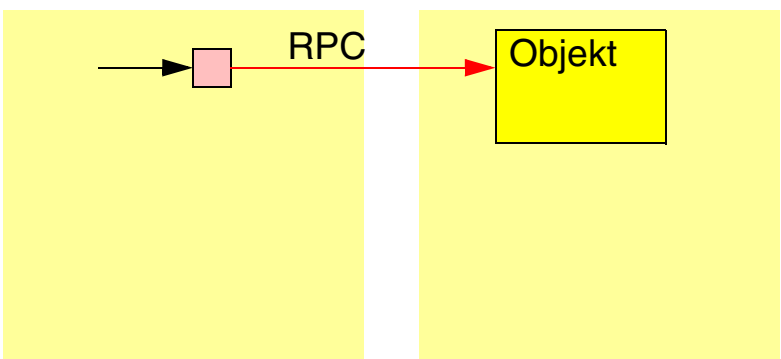
- Middleware-Plattform zur Unterstützung nichtfunktionaler/qualitativer Anwendungseigenschaften
  - Middleware-Kern
  - Framework für fehlertolerante Anwendungen
  - Autonome, mobile Dienste
  - Unterstützung der Anwendungsentwicklung

  - ein wissenschaftlicher Mitarbeiter +
    zwei wissenschaftliche Mitarbeiter am Lehrstuhl Verteilte Systeme und Betriebssysteme in Erlangen (Prof. Schröder-Preikschat)
  - DFG-Förderung

2 AspectIX Middleware-Plattform

2.1 Middleware-Kern

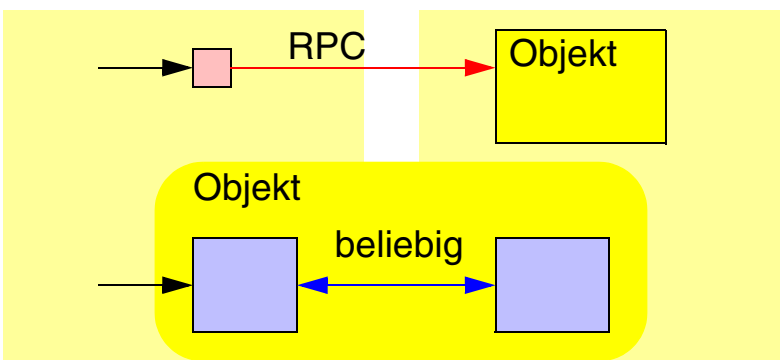
- Fragmentiertes Objektmodell
  - Idee: objektspezifische Fragmente eines verteilten Objekt „vor Ort“
  - transparentes Instanziieren der Fragmente
  - transparente Aufrufe des Objekts



2 AspectIX Middleware-Plattform

2.1 Middleware-Kern

- Fragmentiertes Objektmodell
  - Idee: objektspezifische Fragmente eines verteilten Objekt „vor Ort“
  - transparentes Instanziieren der Fragmente
  - transparente Aufrufe des Objekts



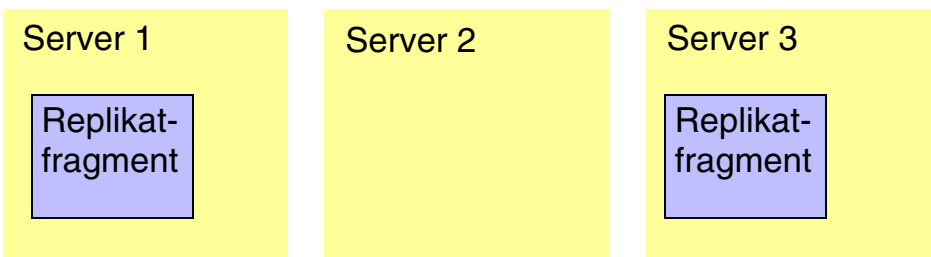
  - Implementierungen: CORBA, Java RMI

2.2 Framework für fehlertolerante Anwendungen

- Zielsetzung
  - Fehlertoleranz durch aktive Replikation
  - bedarfsgerechte Konfigurierbarkeit
    - z.B. geringe Kosten vs. Qualitätseigenschaften
  - dynamisch wechselnde Anforderungen
    - z.B. byzantinische Fehlertoleranz vs. Fail-Stop-Fehlertoleranz
- Framework
  - Satz von Algorithmen zur Einigung (z.B. Paxos, Castro)
  - Kommunikationsschicht zur Kapselung der Algorithmen
  - Integration nahtloser und transparenter Algorithmenwechsel

2.3 Autonome, mobile Dienste

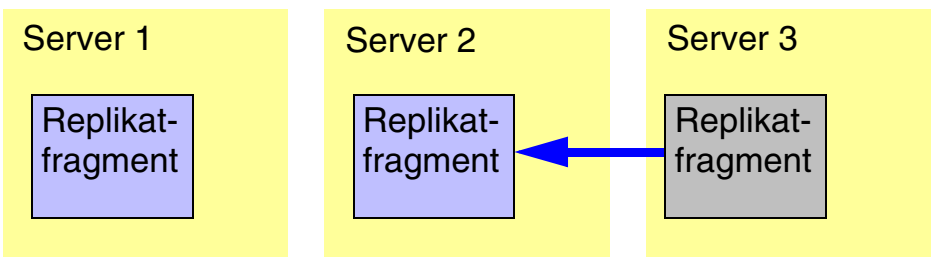
- Zielsetzung
  - Dienstimplementierung durch zustandsbehaftete fragmentierte Objekte
  - automatische Verteilung von Fragmentinstanzen auf Serverrechner
    - verteiltes Objekt ist entscheidende Instanz (Autonomie)



- Infrastruktur
  - „Behausungsdienst“
  - Ressourcenkontrolle
  - Konfigurationswerkzeuge

2.3 Autonome, mobile Dienste

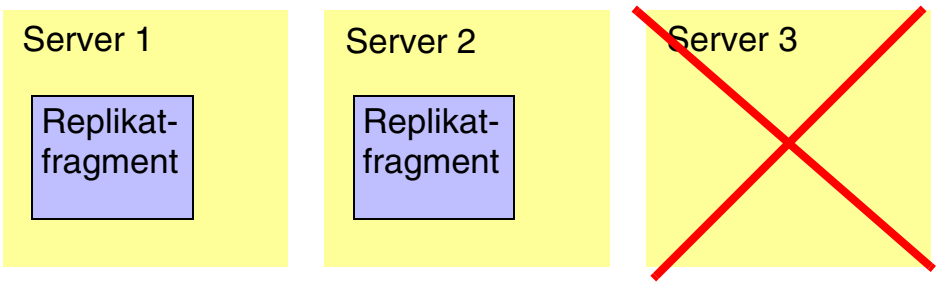
- Zielsetzung
  - Dienstimplementierung durch zustandsbehaftete fragmentierte Objekte
  - automatische Verteilung von Fragmentinstanzen auf Serverrechner
    - verteiltes Objekt ist entscheidende Instanz (Autonomie)



- Infrastruktur
  - „Behausungsdienst“
  - Ressourcenkontrolle
  - Konfigurationswerkzeuge

2.3 Autonome, mobile Dienste

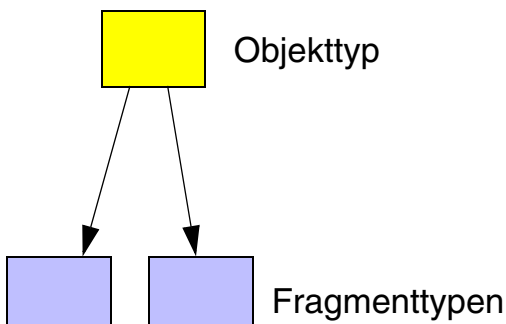
- Zielsetzung
  - Dienstimplementierung durch zustandsbehaftete fragmentierte Objekte
  - automatische Verteilung von Fragmentinstanzen auf Serverrechner
    - verteiltes Objekt ist entscheidende Instanz (Autonomie)



- Infrastruktur
  - „Behausungsdienst“
  - Ressourcenkontrolle
  - Konfigurationswerkzeuge

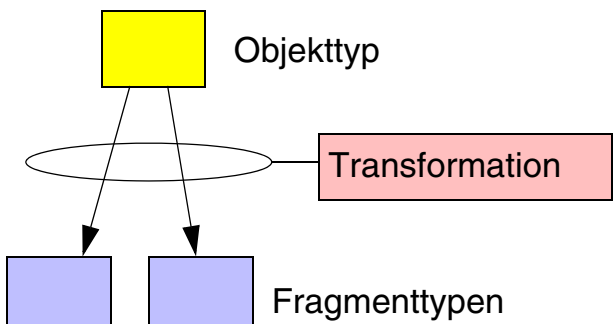
2.4 Unterstützung der Anwendungsentwicklung

- Zielsetzung
  - Code-Generierung für Fragmenttypen
- Transformationsprozesse



2.4 Unterstützung der Anwendungsentwicklung

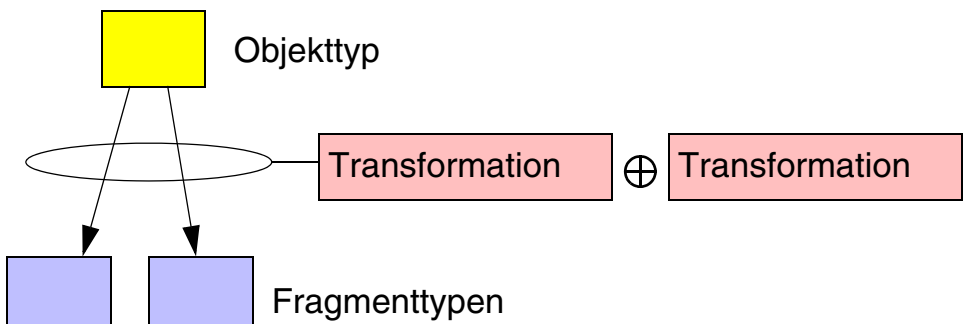
- Zielsetzung
  - Code-Generierung für Fragmenttypen
- Transformationsprozesse



  - Black-Box-Ansatz

2.4 Unterstützung der Anwendungsentwicklung

- Zielsetzung
  - Code-Generierung für Fragmenttypen
- Transformationsprozesse



  - Black-Box-Ansatz
  - Komponierbarkeit der Transformationsprozesse

2.5 Weiteres künftiges Thema

- Schlanke und effiziente Middleware
  - (Soft-)Real-Time
  - modulare, minimale und schnelle Middleware
  - Ressourcen-Kontrolle für Fragmente
- ★ Offene Stelle vorhanden (!)