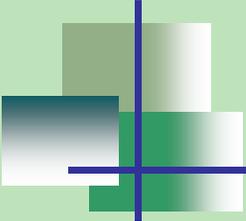
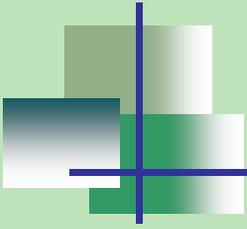


Zwischenbericht Diplomarbeit

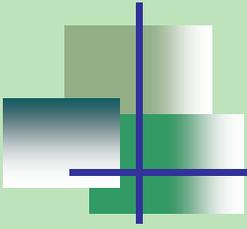


Entwicklung einer
Laufzeitumgebung für
Komponenten mit
Ressourcenanforderungen



Überblick

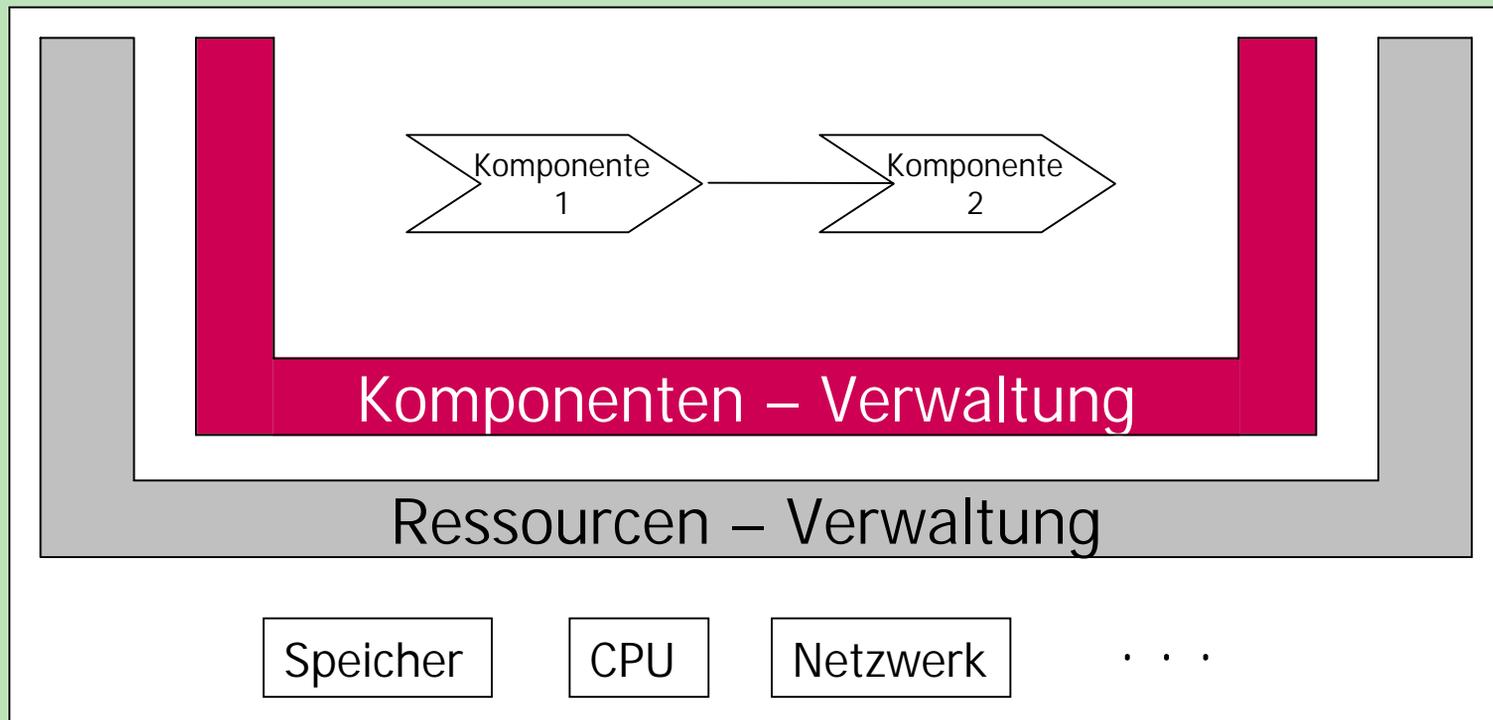
- Beschreibung Aufgabenstellung
- Entwurf der Komponenten – Verwaltung
- Funktionsbereiche
- Zusammenfassung



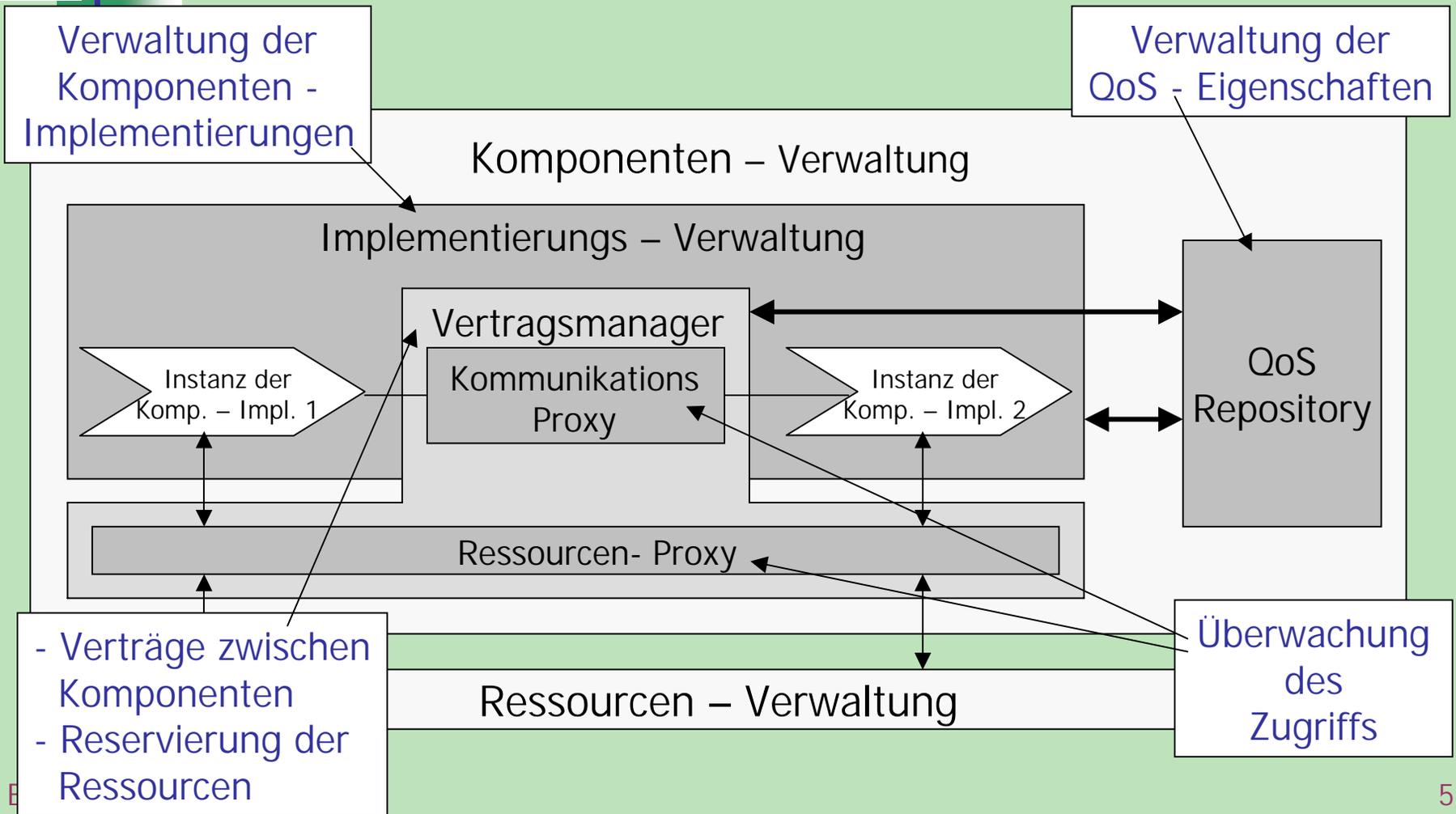
Aufgabenbeschreibung

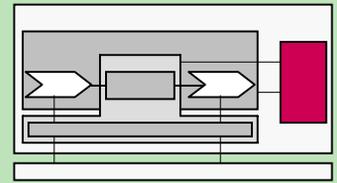
- Anwendungen mit QoS – Anforderungen
 - Spezifikation der Anforderungen mit CQML
(**C**omponent **Q**uality **M**odelling **L**anguage)
- Container zur Durchsetzung der QoS -
Eigenschaften

Aufgabenbeschreibung



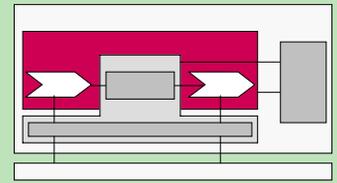
Komponenten - Verwaltung





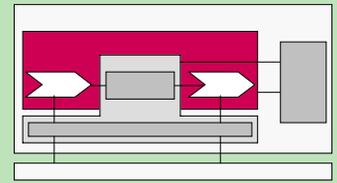
QoS – Repository

- Aufgaben:
 - Verwaltung der QoS – Eigenschaften
 - Zusammenarbeit mit allen Funktionsbereichen der Komponenten – Verwaltung
- Erstellter Entwurf:
 - Abbildung der CQML – Spezifikationen
 - Bindung der QoS – Eigenschaften an Komponenten
 - Auswertung der Eigenschaften bei Aushandlung der Verträge



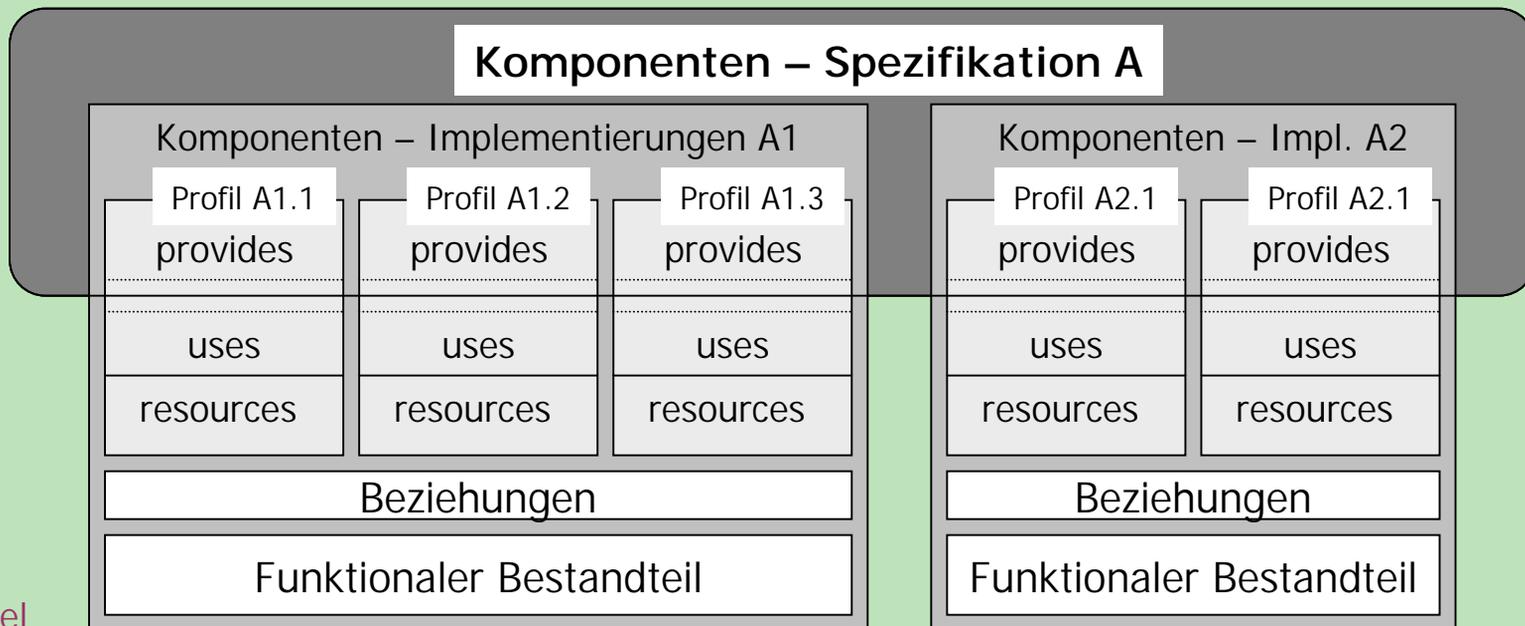
Implementierungs – Verwaltung

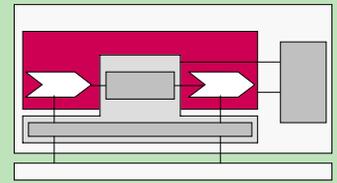
- Verwalten von Komponenten – Implementierungen
 - Einfügen und Entfernen
 - inkl. Beziehungen zwischen Komponenten
- Verwalten der Instanzen



Implementierungs – Verwaltung

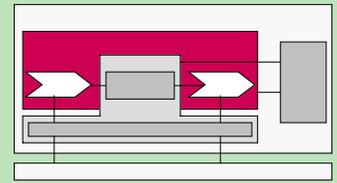
- Komponenten – Spezifikationen
 - Komponenten – Spezifikation = Schnittstellen
 - Zuordnung mehrerer Komponenten – Implementierungen mit unterschiedlichen QoS - Eigenschaften



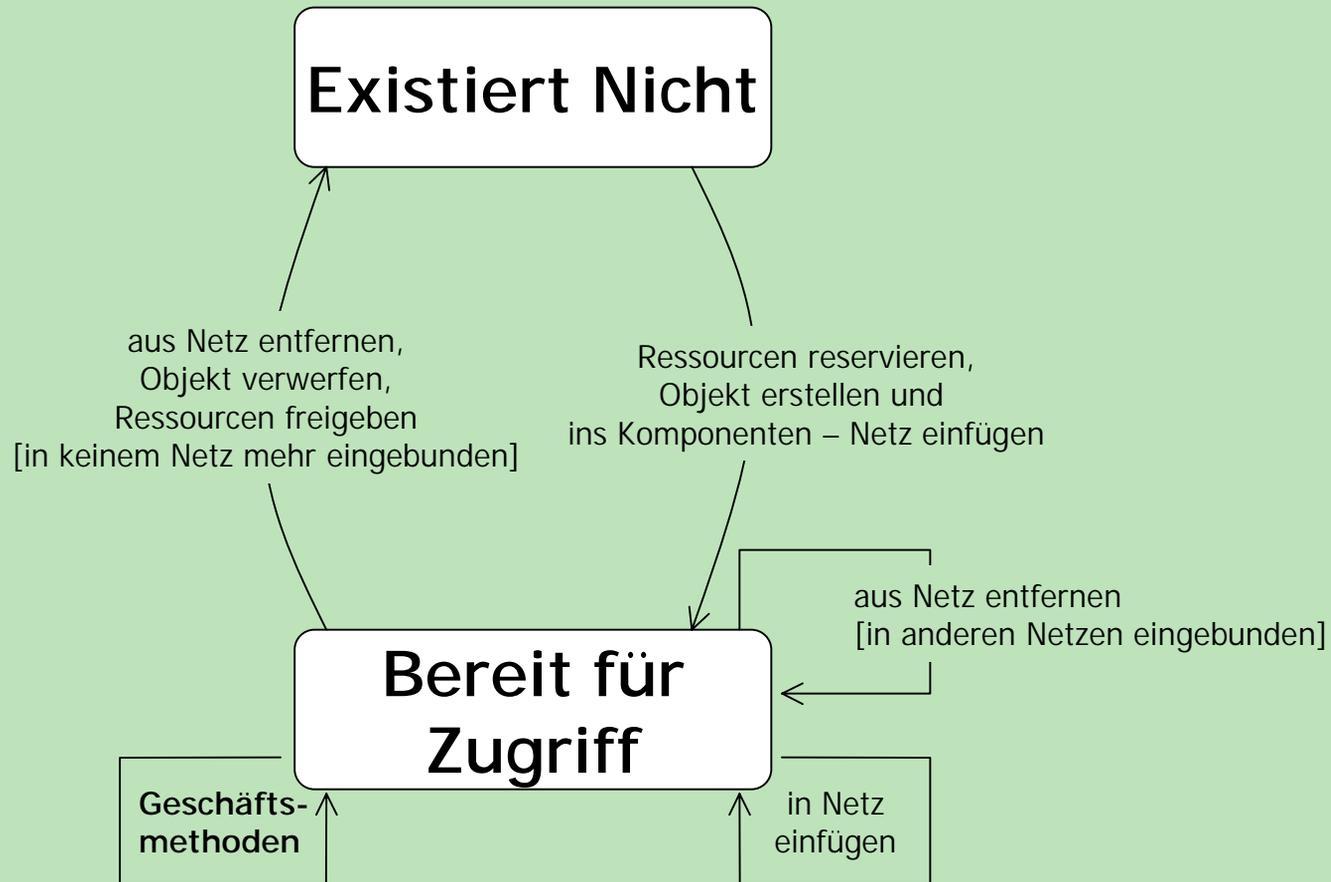


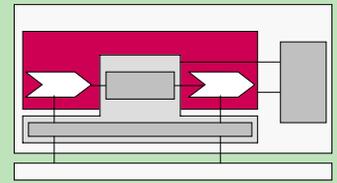
Instanzen – Verwaltung

- Verarbeitung unterschiedlicher Komponententypen
- Orientierung an EJB
 - *Stateful Session Beans*
 - *Stateless Session Beans*
 - *Entity Beans*
 - keine Verarbeitung von QoS – Anforderungen

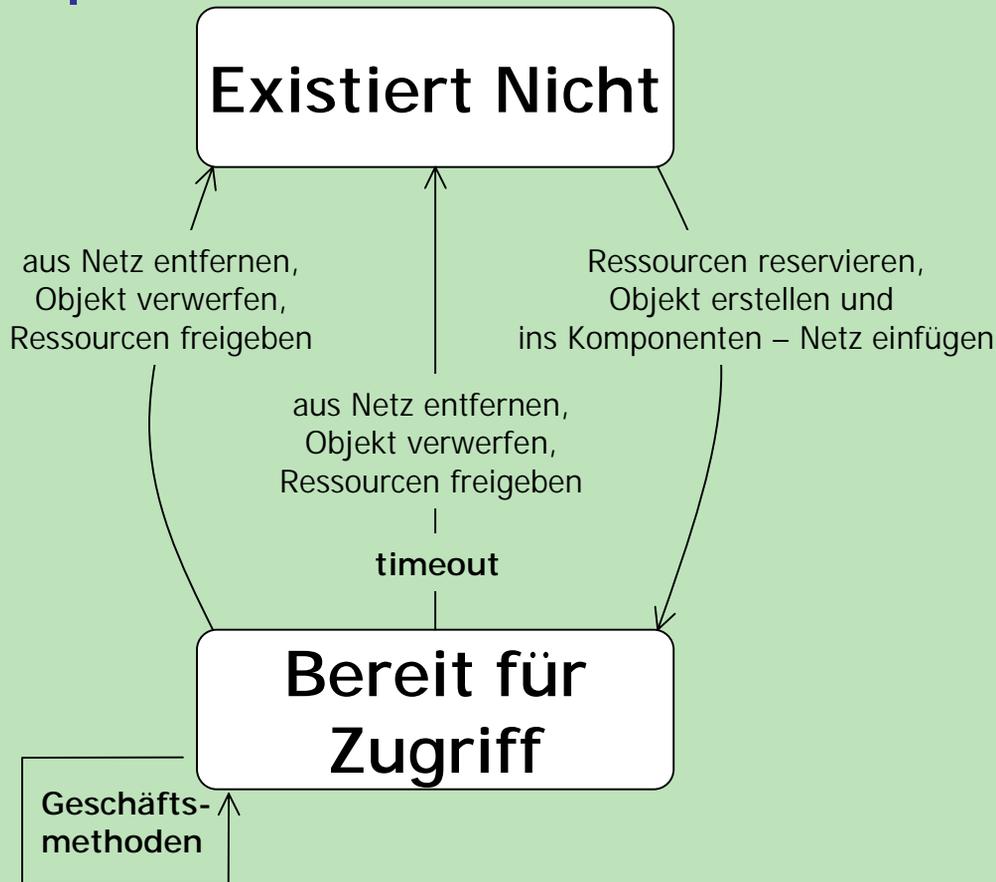


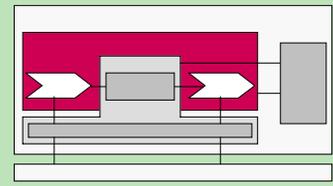
Lebenszyklus: *Stateless Session Beans*



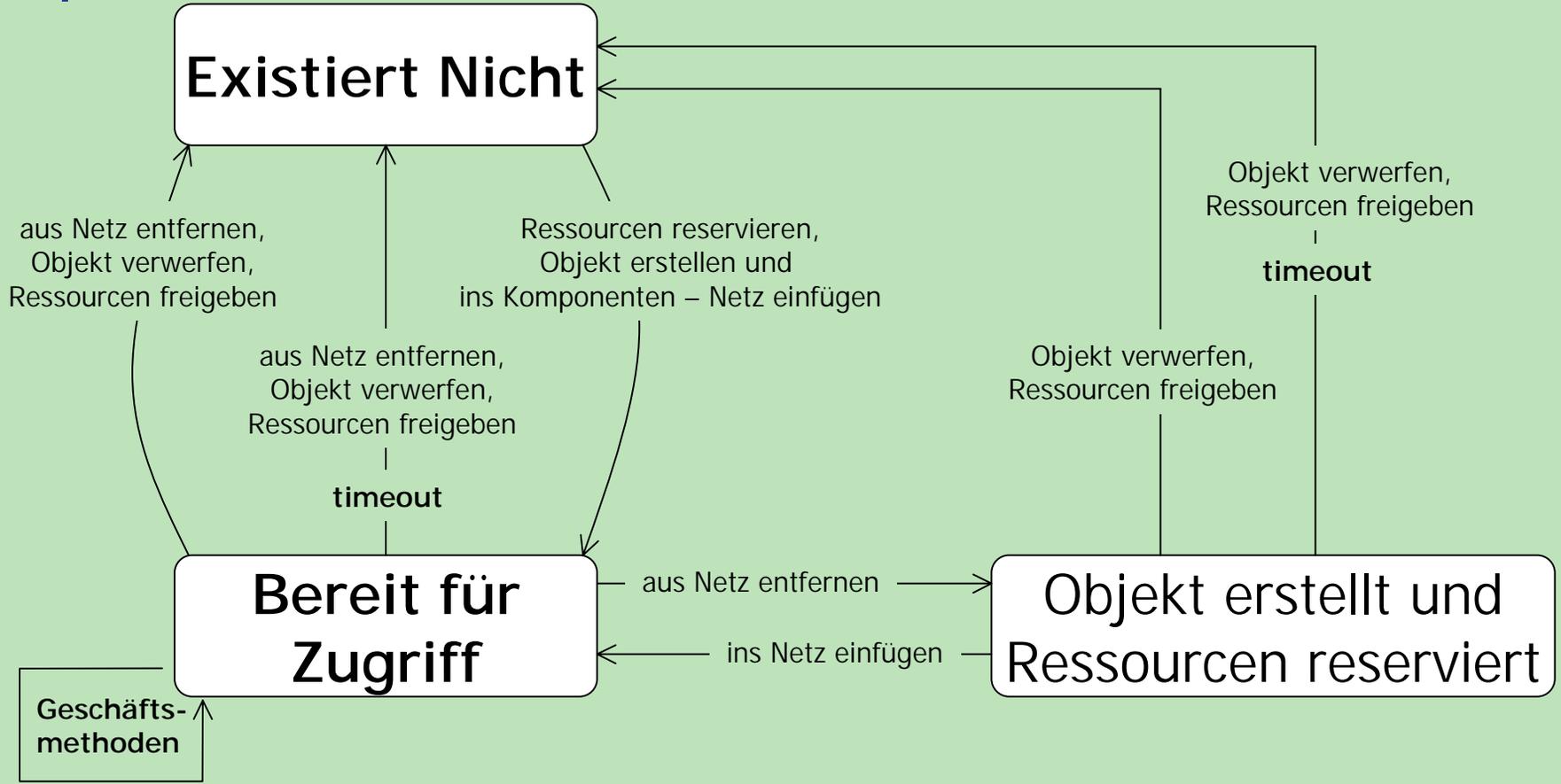


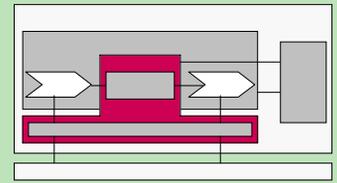
Lebenszyklus: *Stateful Session Beans*





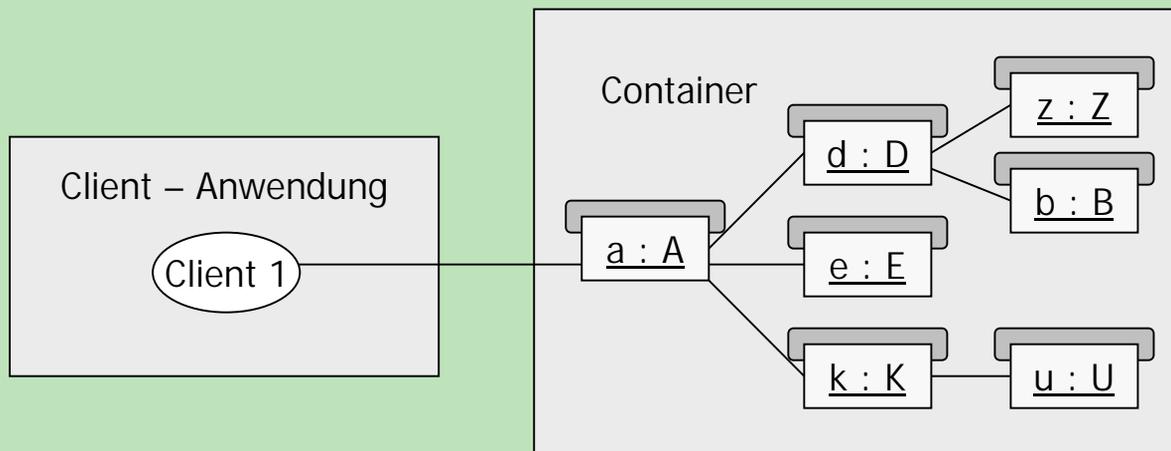
Lebenszyklus: *Stateful Session Beans*

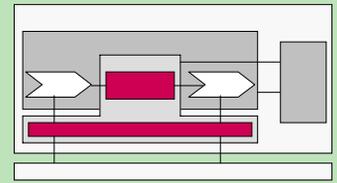




Vertragsmanager I.

- Aufruf vom Client (mit QoS – Anforderungen)
 - Aushandlung der Verträge:
 - QoS – Eigenschaften zwischen kommunizierenden Komponenten
 - Reservierung von Ressourcen
- Erstellung des Komponenten – Netzes

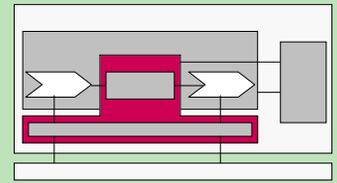




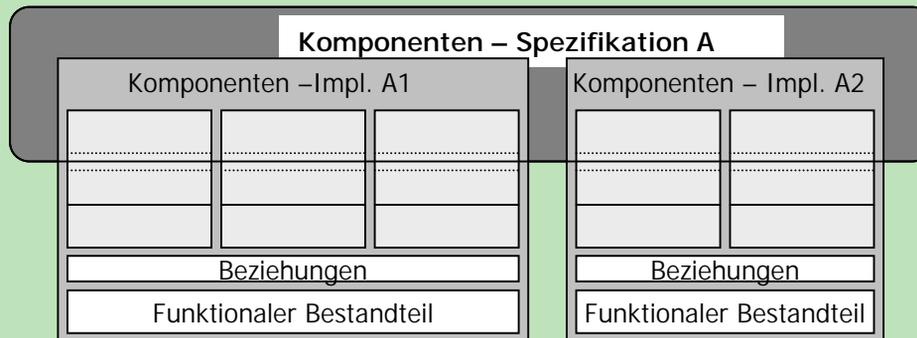
Vertragsmanager II.

- Ausführung von Clientanfragen (Durchsetzung Verträge)
 - Zuweisung reservierter Instanzen (Kommunikations – Proxy)
 - Zugriff auf reservierte Ressourcen (Ressourcen – Proxy)

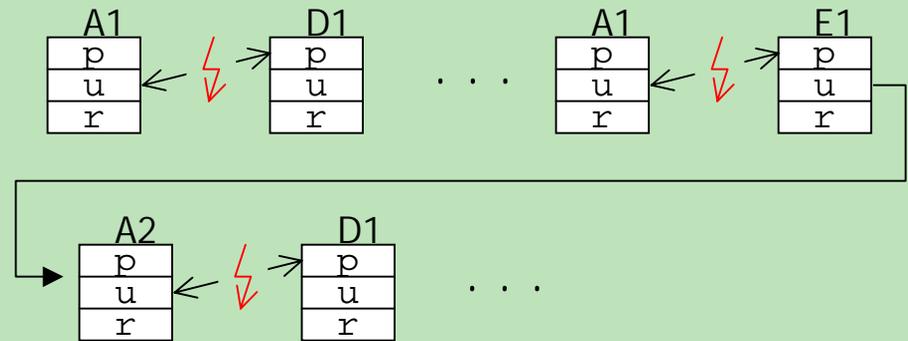
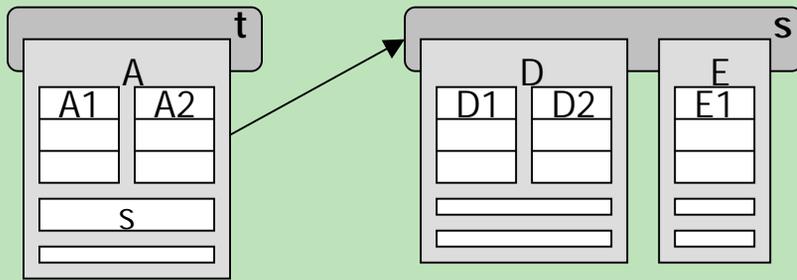
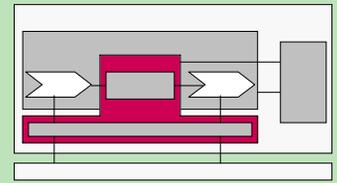
Vertragsaushandlung: zwischen Komponenten



- Beziehungen = verwendete Schnittstellen

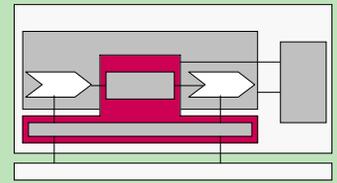


Vertragsaushandlung: zwischen Komponenten



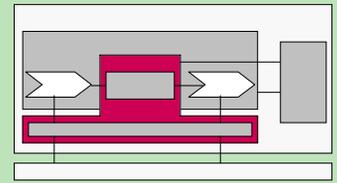
- Vergleich der Profile
 - uses mit provides der verwendeten Implementierung
 - Reservierung der Ressourcen resources
- Suche bis passendes Profil gefunden

Vertragsaushandlung: zwischen Komponenten

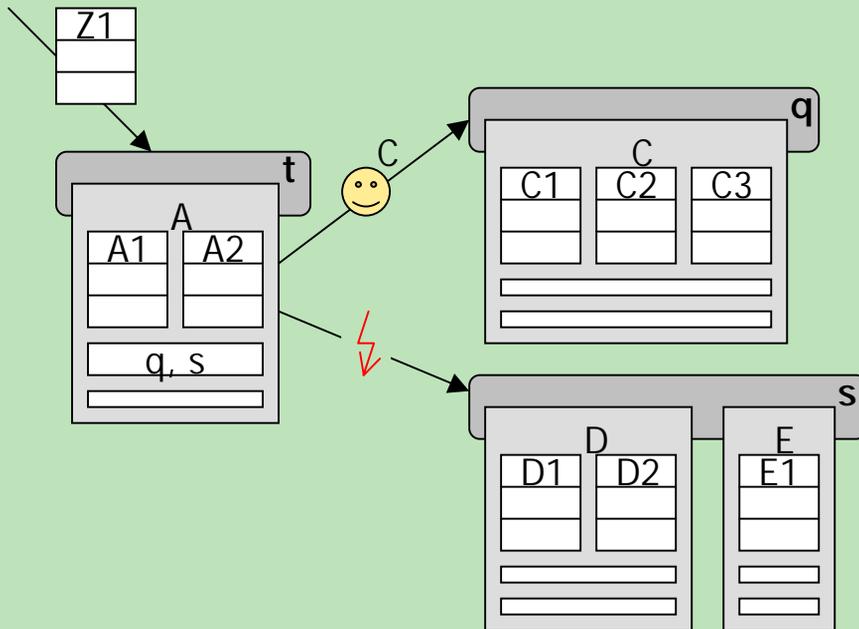


- Suche scheitert:
 - im Netz schrittweise zurück gehen bis
 1. in verwendeter Implementierung alternatives Profil oder
 2. in Spezifikation alternative Implementierung
 - bei jedem Schritt zurück – Freigabe der Ressourcen
 - Alternatives Profil gefunden:
 - Vollständig neue Reservierung abhängiger Schnittstellen

Vertragsaushandlung: zwischen Komponenten

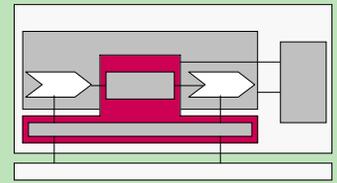


Beispiel: Alternatives Profil

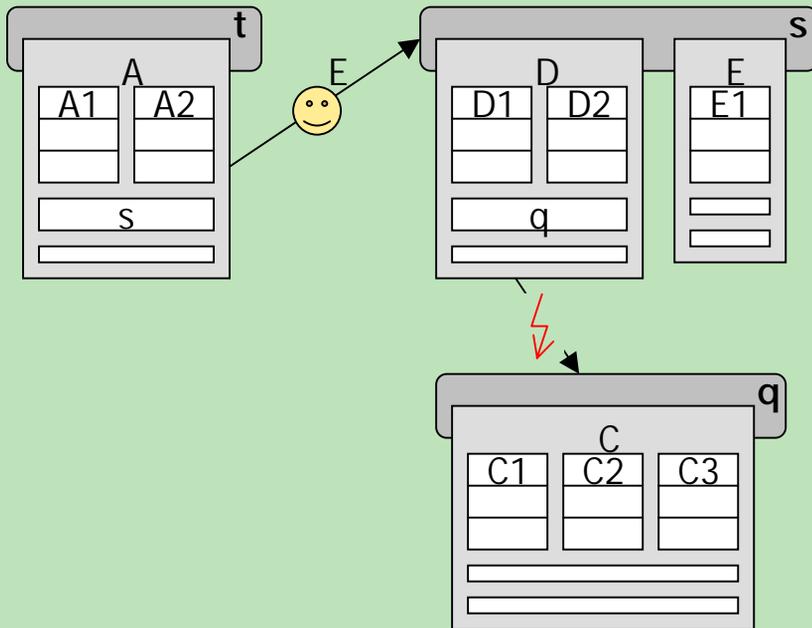


- c erfüllt Anforderungen von A1
- keine für Profil A1 passende Komponenten – Implementierung in s
- Freigabe von c
- Erneuter Versuch mit Profil A2

Vertragsaushandlung: zwischen Komponenten

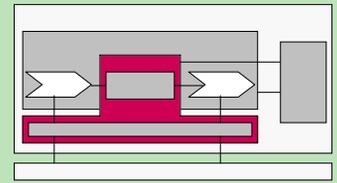


Beispiel: Alternative Implementierung



- D erfüllt Anforderungen von A
- D findet keine passende Implementierung in q
- Freigabe D
- Test mit \mathbb{E} erfolgreich

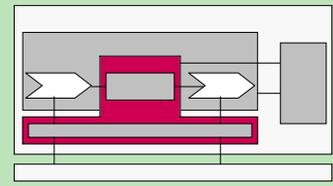
Vertragsaushandlung: mit Ressourcen - Verwaltung



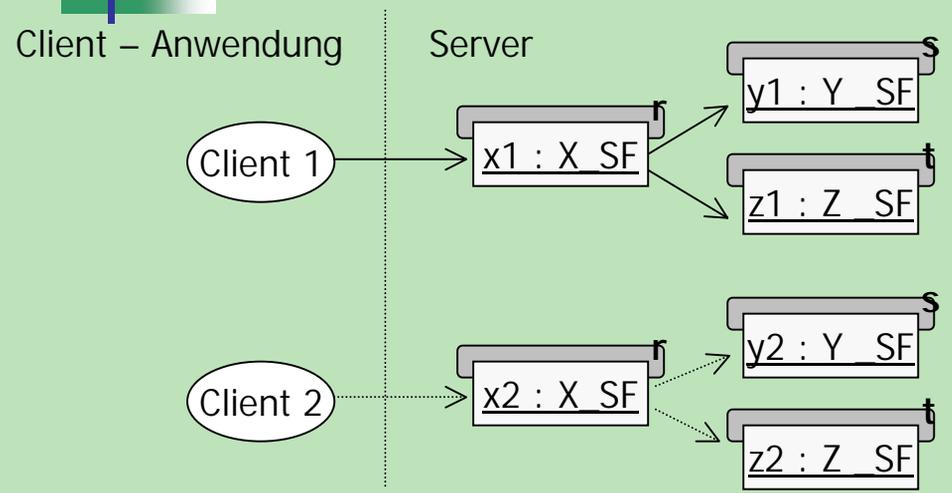
- Reservierung der Ressourcen
 - Spezifikation der Anforderungen in `resources`
 - Anfrage an Ressourcen – Verwaltung
 - Reservierung der Ressourcen
 - Rückgabe Handle zum Zugriff auf reservierte Ressourcen
 - Zuordnung der Handle zu den Instanzen
 - Instanzerstellung – Verwendung der Ressourcen!

Handle:

- Verweis auf verwendete Ressourcen – Verwaltung
- Beschreibung der reservierten Ressource

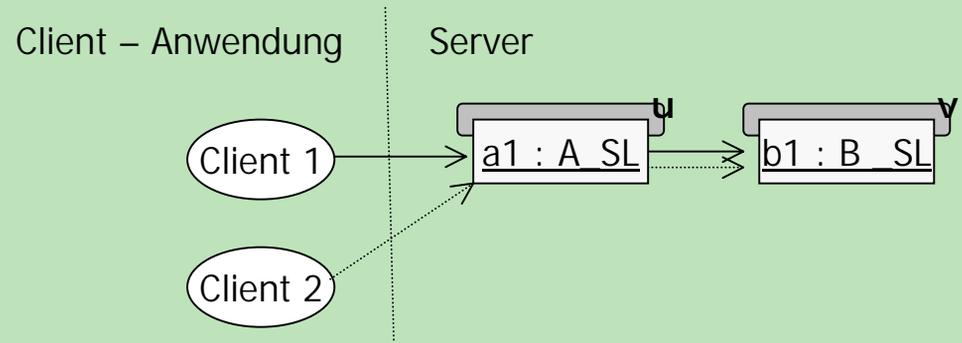


Komponenten – Netze



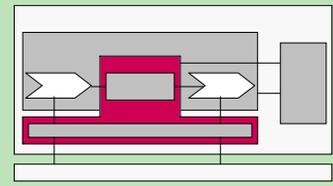
nur *Stateful Session Beans*

- jeder Client eigene Instanzen
- bei großen Netzen Konversationszustand für langen Zeitraum zu halten
- timeout einer Instanz:
 - Verlust Konversationszustand
 - Netz unbrauchbar



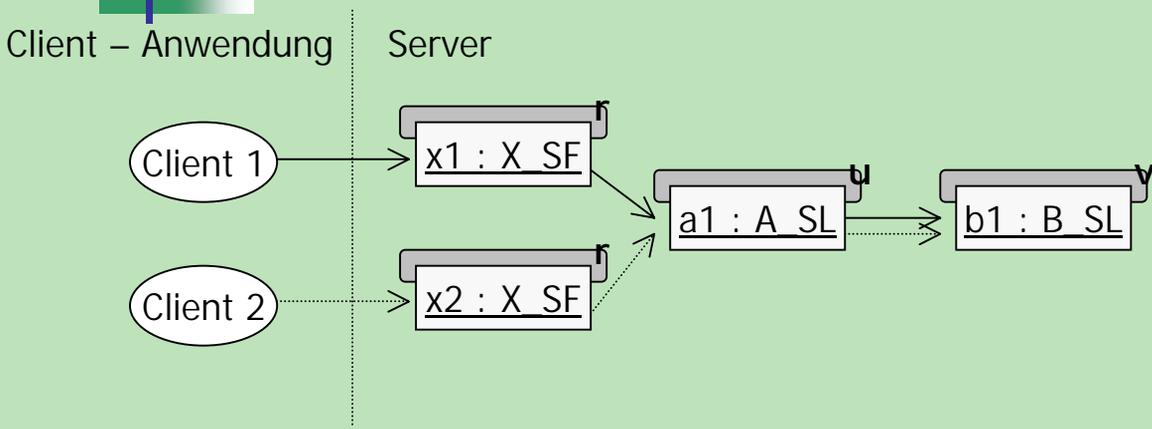
nur *Stateless Session Beans*

- unproblematisch



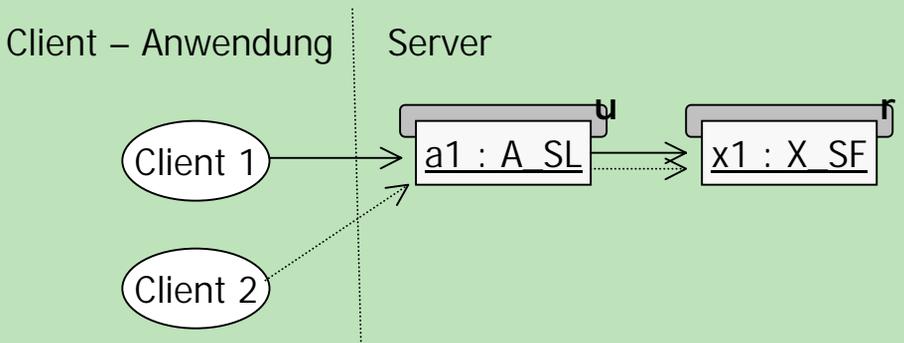
Komponenten – Netze

Kombination von *Stateless* und *Stateful Session Beans*



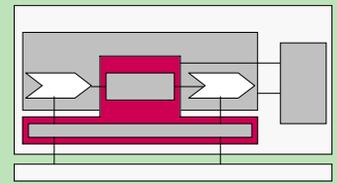
Stateful → *Stateless*

- unproblematisch



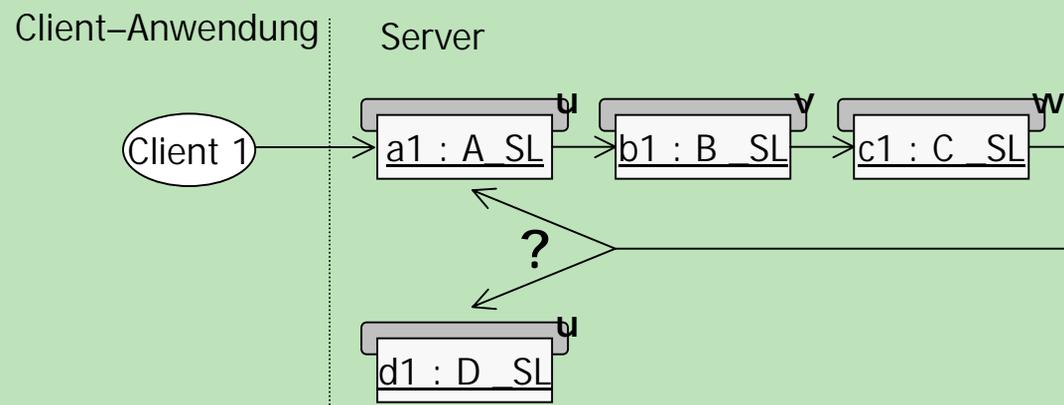
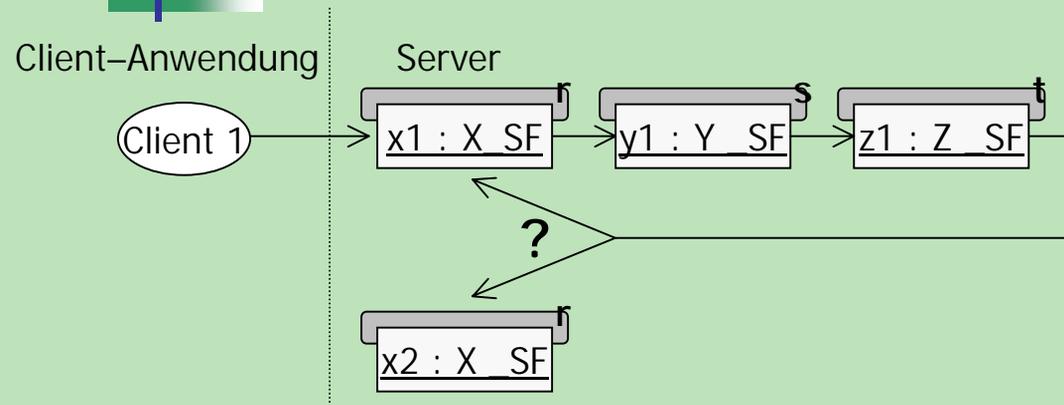
Stateless → *Stateful*

- eigentlich unsinnig:
Konversationszustand verloren



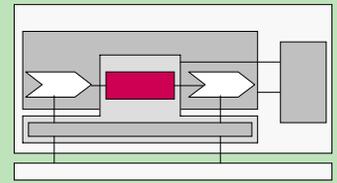
Komponenten – Netze

Schleifenbildung



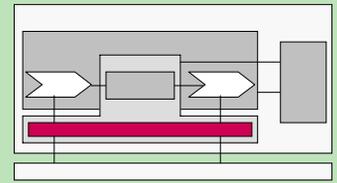
- Abbruch Reservierung
 - keine weitere Abhängigkeiten oder
 - bei Verweis auf bereits eingebundene Implementierung
- Zugriff auf existierende Instanz oder neue erstellen?
- Spezifikation dieser Beziehungen
 - ADL ?

Vertragsdurchsetzung: Kommunikations – Proxy



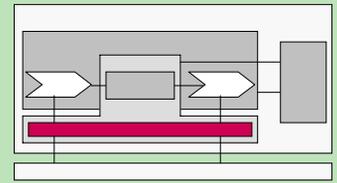
- Aktivierung:
 - Anforderung einer Instanz zu einer Schnittstelle
 - Methoden eingeführt:
 - `getLocalHome(String interface)`
 - `getRemoteHome(String interface)`
- Maßnahme:
 - Rückgabe der reservierten Instanz
- Ausnahmesituationen:
 - Netzerweiterung
 - Mehrfacher `create` Aufruf
 - Austausch von Netzen und Teilnetzen

Vertragsdurchsetzung: Ressourcen – Proxy

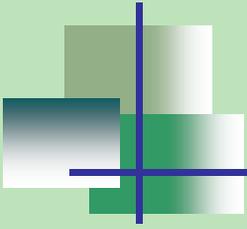


- Aufgaben:
 - Durchsetzung der Verträge mit der Ressourcen – Verwaltung
 - Aktivierung bei Zugriff auf Ressourcen
 - Zuweisung reservierter Ressourcen
 - Überwachung der Einhaltung der vereinbarten Verträge (z.B. Speichermenge)
 - durch Ressourcen – Verwaltung
- Ressourcen
 - CPU, Speicher, Netzwerk

Vertragsdurchsetzung: Ressourcen – Proxy

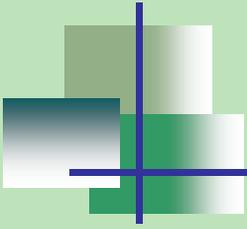


- Lösungsansätze:
 - CPU
 - keine Überwachung
 - Aktivierung bei Zugriff:
 - Speicher
 - Konstruktor – Ausführung (`new`)
 - Freigabe der Ressourcen: Destruktor
 - Netzwerk
 - Aufruf `getRemoteHome`
 - Zuweisung reservierter Ressourcen:
 - Verweis auf Ressourcen – Verwaltung gespeichert
 - Zugriff auf Speicher ein Problem in Java!
 - Konstruktor überschreiben
 - Verwendung von JNI (C, C++)



Zusammenfassung

- Entwurf fast abgeschlossen:
 - Instanzen – Verwaltung
 - Verwaltung QoS – Eigenschaften
 - Kommunikations – Proxy
- Unvollständiger Entwurf:
 - Vertrags – Aushandlung (ADL ?)
 - Zuweisung der Ressourcen (Ressourcen – Proxy)
- Verbleibende Aufgaben:
 - Abschluss Entwurf
 - Implementierung des Prototypen



Diskussion

- Vertrags – Aushandlung
 - Abhängigkeiten zwischen Instanzen definieren

- Ressourcen – Proxy:
Zuweisung der Ressourcen
 - Ideen zum Zugriff auf Ressourcen