

Autonome Koordination verteilter Services – Ein neues Paradigma und seine Bedeutung für E-Government

A.o. Univ. Prof. Dr. DI eva Kühn
TU Wien, Institut für Computersprachen
Space Based Computing Group
1040 Wien, Argentinierstr. 8
www.complang.tuwien.ac.at/eva
eva.kuehn@tuwien.ac.at

..... FAKULTÄT FÜR INFORMATIK

Service Oriented Architectures: “Decoupling” und Choreographie

- Service Orientierte Architekturen (SOA)
 - Sind der unaufhaltbare Trend in IT-Architekturen
 - According to WinterGreen Research, the SOA market will grow from \$450 million in 2005 to reach \$18.4 billion by 2012.
- “Decoupling”
 - Unabhängigkeit von Service Consumer und Service Provider
 - Plattform
 - Lokation
 - Verfügbarkeit
 - etc.
- Choreographie
 - Zusammenspiel von Services nach vordefinierten Regeln
 - Flexibel Anwendungen aus bestehenden Services bauen
- Neue Herausforderung: Autonomie
 - Choreographie ohne Bindung an vorbestimmtes, „allwissendes“, fix-verdrahtetes Skript, das ein Koordinator exekutiert, von dem alles abhängt

..... FAKULTÄT FÜR INFORMATIK

- Services werden von Clients aufgerufen
- Service wird nur auf **Aufforderung** hin aktiv



- Verbindungen (Medium) zwischen den Services: **Gerichtete Kanäle**
 - „FIFO“ (first-in / first-out)
- Bewertung:
 - Zentraler allwissender Koordinator
 - Für asymmetrische Verhältnisse – unterschiedliche Partner
 - Nachrichten werden konsumiert
 - Service kann sich nicht an Gegebenheiten anpassen
 - Umgang mit ungeplanten Ereignissen schwierig
 - Wenig Interaktionsmuster: **in/out („FIFO“)**

..... FAKULTÄT FÜR INFORMATIK

- „Zuruf“ ist falsche Metapher für **autonome** Geschäftspartner
- Wie wollen wir **zusammenarbeiten**?



- Autonome Services, die selbständig und symmetrisch agieren
- Gemeinsam** und organisationsübergreifend ein Ziel erreichen
- Man benötigt mehr Info als nur aus dem „Schlüsselloch“ seiner Pipe
- **Gemeinsamer, zustandsbehafteter „Datenraum“**

..... FAKULTÄT FÜR INFORMATIK

■ Inversion der Kontrolle

- 1) **Ich** beobachte etwas
- 2) **Ich** hole mir Information
- 3) **Ich** tue etwas

■ Autonome Service Orientierte Architekturen

- Ob & wann
 - Dynamisches Kommen & Gehen
 - Sich selber orientieren / koordinieren
 - Kein „allwissender“ Koordinator/Vordenker
 - Zerlegung in kleinere, autonome, überschaubarere Einheiten
 - z.B: Wasserrohrbruch am Wochenende
 - Ich bestimme ob & was ich tue
 - Nicht nur FIFO ...
- Skalierbarkeit**
Komplexitätsreduktion
Nicht planbare Situationen
Sicherheit
Beliebige Kooperationsformen

..... FAKULTÄT FÜR INFORMATIK

Einige Services:

■ Service 1: „Tisch zugewiesen bekommen“

Client: Gast
Server: Kellner
Medium: FIFO ✓

FIFO



■ Service 2: „Bestellung in der Küche“

Client: Kellner
Server: Koch
Medium:
 • FIFO ✗
 Küche arbeitet ineffizient
 Unvorhersehbarer, unplanbarer Ereignisablauf
 • Gemeinsames Whiteboard ✓

Whiteboard



■ Service 3: „Essen am Tisch serviert bekommen“

Client: Gast
Server: Kellner
Medium:
 • FIFO ✗
 Kellner stapelt alle Speisen in einer „Röhre“ vorm Gast
 Gast muß sie in dieser Reihenfolge konsumieren
 • Gemeinsamer strukturierter Tisch ✓
 Freier, koordinierter Zugriff

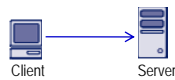


Tisch

■ Anforderungen:

- Ich** beobachte etwas → Über Ereignisse informiert werden (1)
- Ich** hole mir Information → Selektiver Zugriff auf strukturierte Daten (2)
- Ich** tue etwas → Service Implementierung ✓

(1) flüchtige, schnelle Notifikation



Fix-verdrahteter Aufruf
z.B: RPC mit SOAP
(1) JA aber keine Entkoppelung
(2) NEIN

(1-2) temporäre, Zwischen-
bufferung



Message Queue
(1) JA
(2) NEIN

(2) persistente, strukturierte, große Daten



Datenbank
(1) NEIN
(2) JA

■ Anforderungen:

- Ich** beobachte etwas → Über Ereignisse informiert werden (1)
- Ich** hole mir Information → Selektiver Zugriff auf strukturierte Daten (2)
- Ich** tue etwas → Service Implementierung ✓

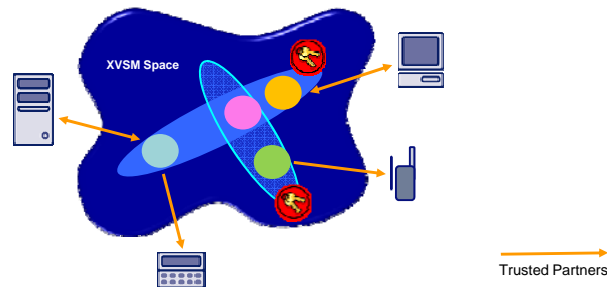
(1) flüchtige, schnelle Notifikation



(1-2) temporäre, Zwischen-
bufferung

„Space“
(1)+(2) ok

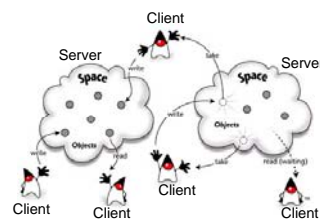
(2) persistente, strukturierte, große Daten



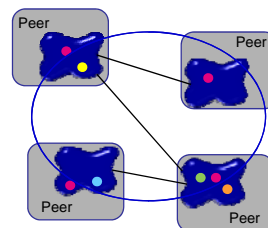
- Gleichberechtigte Service-Peer interagieren in verschiedenen Rollen durch **Lesen und Schreiben von strukturierten Daten** in einem sicheren, gemeinsamen, virtuellen Datenraum
- Notifikation über Änderungen in Quasi-Echtzeit
- **Koordination über gemeinsame Datenstrukturen** statt Kommunikation!

..... FAKULTÄT FÜR **INFORMATIK**

- **Tuple Space**
 - JavaSpaces (SUN)
 - T-Spaces (IBM)
 - XML Spaces (Univ. Berlin)
 - etc.



- **Extensible Virtual Shared Memory (TU Wien)**
 - Erweitert bestehende Ansätze
 - Echte Peer-to-Peer Architektur
 - Einfaches API: read, write, take, notify
 - Internetfähiges XML Protokoll
 - Soll offener Standard werden
 - Open Source: www.xvsm.org



- „Real-Time Collaboration“:
 - Definition: kein zentraler Koordinator, on-line, near-time, gemeinsam
 - Whiteboard ist effektiver als Email
 - Vorboten: Instant Messaging, Skype, Web 2.0 Applikationen
- **Nicht nur Menschen, auch Software-Services wollen so kooperieren!**
- Beides ist möglich mit Spaces:
 - push \longleftrightarrow pull
 - in-memory \longleftrightarrow persistent
 - lokal \longleftrightarrow remote
 - FIFO \longleftrightarrow beliebig
 - zentral \longleftrightarrow verteilt
 - single-copy \longleftrightarrow repliziert
- Immer in Kombination mit
 - Vereinheitlichten Orthogonale Services (Monitoring, Auditing, Security etc.)

..... FAKULTÄT FÜR INFORMATIK

- Nicht vorausplanbare Situationen
 - Schneller reagieren können (Krisensituationen)
- Mobile Geräte
 - Zugriff von überall
- Große Anzahl von Services
 - Heterogenität und hohe Änderungsdynamik (Dienstleistungsrichtlinie)
- Große Anzahl von Benutzern
 - Peer-to-Peer: Skalierbarkeit und Ausfallsicherheit
- Neue Kollaborative Enterprise Services („Web 2.0“)
 - on-line Partizipation in Entscheidungen (E-Procurement)
- Weitere Visionen aus der Forschung
 - Spaces mit Semantik
 - Spaces mit Selbstorganisation

..... FAKULTÄT FÜR INFORMATIK