JIT-Konferenz - 20. September 1999

Caching in Stubs und Events mit EJB bei Einsatz einer OODB

Dipl.-Inform. Olaf Neumann
cand. Dipl.-Inform. Christoph Pohl
Dipl.-Inform. Katrin Franze

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - Inhalt

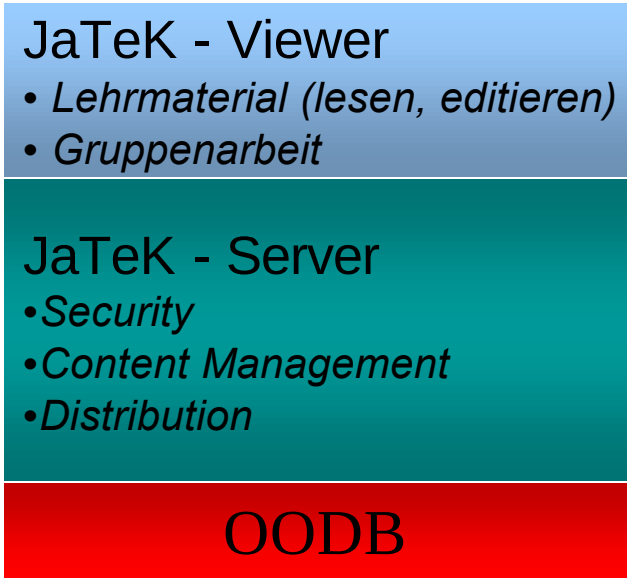
- JaTeK - Ausgangssituation/Szenarien
- Anforderungen
- Einsatz von Enterprise JavaBeans (EJB)
- Einbindung einer objektorientierten Datenbank
- Erfahrungen
- Ausblick

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - Szenario

JaTeK - Teleteaching/Telelearning Plattform

- statisch: Dokumente, Aufgaben, ...
- dynamisch: Gruppenarbeit, ...

- proprietärer Server
- Einsatz einer OODB

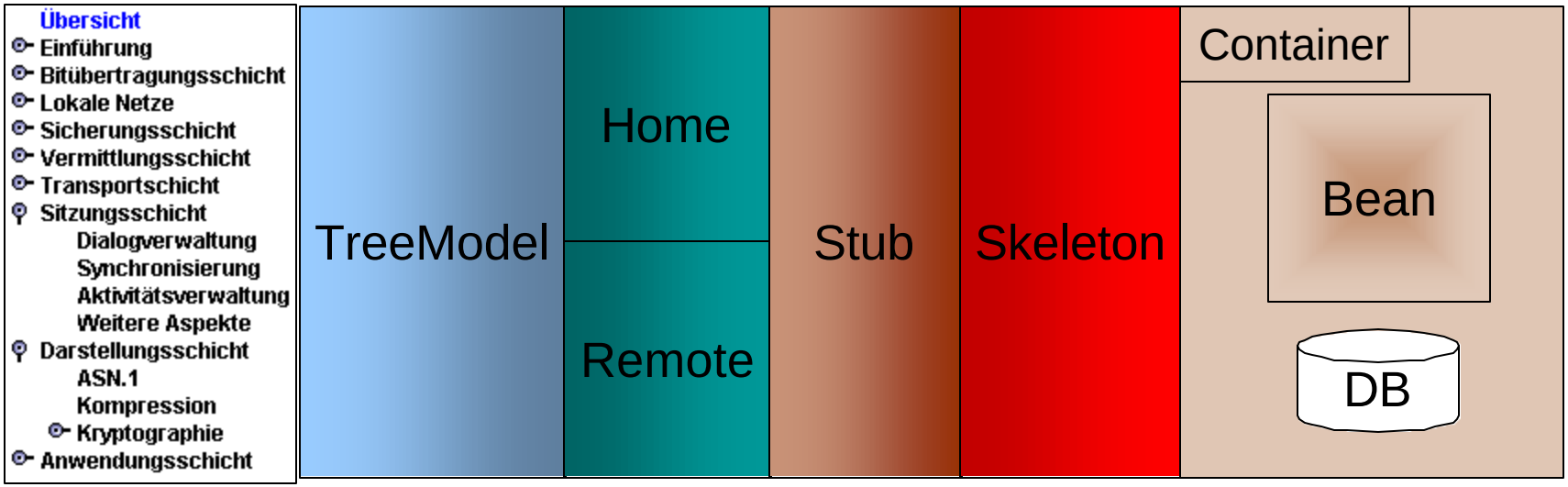


Caching in Stubs und Events mit EJB bei Einsatz einer OODB - Anforderungen

- Zentrierter Server:
  Übergang von proprietärer Lösung zu EJB
  Security auf Dokument- oder Objektebene;
  objektorientierte Datenbank
- Verteilung dynamischer Daten:
  RMI-Callback (garbage collection)
  JSDT (Sessions, Channel)
  JINI (JavaSpaces)
  Corba-Event-Service (Push, Pull)

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - EJB

- Einsatz von EJB:
  Skalierbarkeit, Lastausgleich, Transaktionen
  Standardkonforme Schnittstellen



Caching in Stubs und Events mit EJB bei Einsatz einer OODB - EJB (Überlegungen)

- Aufteilung der Anwendung (Client/Server)
- Einsatz einer objektorientierten Datenbank
- Sicherheit durch Rollen und ACLs
- Performance

⇒eigener Server bzw. Servererweiterung

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - EJB (Überlegungen)

- Aufteilung der Anwendung
  Aufgaben, Experimente $\Rightarrow$ viele Interaktionen
  kein Thin-Client möglich

  wenige Aufrufe mit vielen Daten
  Caching vs. Hauptspeicherbedarf (Proxy)
  Iteratoren
  Aktionen im SessionObjekt zusammenfassen
  GUI Entkopplung (Threads)

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - EJB (Überlegungen)

- Stub und Skeleton werden generiert
  a) geeignete Eigenschaften definieren
  b) Eigenschaften beim Generieren berücksichtigen
  c) asynchrone Events integrieren

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - OODB

- Bean-Attribute ⇒ persistente Attribute (Auswahl)
- Primary key ⇒ einheitliche OID
- finderXXX()-Methoden ⇒ Attribute ⇒ Collections (Erreichbarkeit, Suche)
- create() ⇒ neues persistentes Objekt

- Zusammenarbeit Server/DB vs. extra Klassen ⇒ Schnittstelle in Container/Server für OODB

Caching in Stubs und Events mit EJB bei Einsatz einer OODB -Erfahrungen

- Caching:
  akzeptable Performance
  verteilte Applikationen durch Events
- OODB:
  kein Impedance Mismatch
  50% weniger Quelltext
  geringere Entwicklungszeit
  deutlich weniger fehleranfällig
- Design-Richtlinien für EJB wichtig

Caching in Stubs und Events mit EJB bei Einsatz einer OODB - Ausblick

- Versant Enterprise Container, Objectstore und BEA Weblogic, Iona ...
- Vergleichsmessungen
- JINI-Events (Lightweight Events)
- JMS (Java Messaging Service)
- Attribute in Beans später laden (Aktivierung)