

Java 2, Enterprise Edition Einführung und Überblick

Martin Husemann
m_husema@informatik.uni-kl.de

Vortragsinhalte

- Rahmenbedingungen in der heutigen Marktwirtschaft
- Mehrschichtige Systemarchitekturen
- Komponenten-Architekturen für die Server-Ebene
- Typische Strukturen von J2EE-Systemen
- Technologien und APIs

Rahmenbedingungen

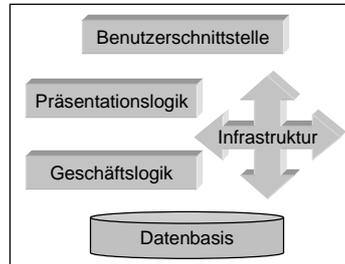
- Information als entscheidender Wettbewerbsfaktor
 - gleichrangig neben Gütern und Dienstleistungen
 - „Informationswirtschaft“
- Einsatz moderner Technologien zwingend nötig
- Integration unterschiedlicher Systeme
- Daten aus heterogenen Quellen

Herausforderungen an Applikationsentwickler

- Kurze Reaktionszeiten
 - angemessene Reaktion auf neue Trends und Technologien
- Produktivität der Programmierung
 - sinnvolle Kombination von Neuem und Bewährtem
- Verfügbarkeit und Zuverlässigkeit
 - keine Ausfallzeiten; im regulären Betrieb zuverlässiges Verhalten
- Sicherheit
 - Schutz kritischer Daten in allen Nutzungsszenarien
- Skalierbarkeit
 - leichtes Wachsen mit steigenden Anforderungen
- Integration
 - Zusammenarbeit mit bestehenden Systemen

Mehrschichtige Systemarchitekturen

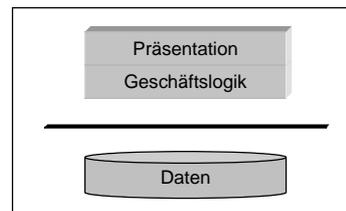
- Benutzerschnittstelle
- Präsentationslogik
- Geschäftslogik
- Datenbasis
- Infrastruktur



- Umsetzung in konkrete Schichten und Ebenen
 - Präsentation
 - Geschäftslogik
 - Daten

Zwei-Ebenen-Architekturen

- Präsentations- und Geschäftslogik auf Client-Ebene
 - Server verwaltet nur Daten
 - hoher Wartungsaufwand
 - schlechte Skalierbarkeit



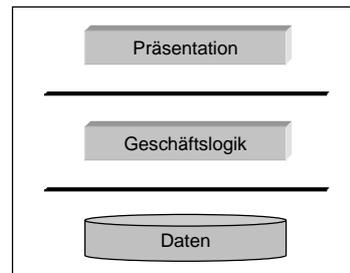
- Teile der Geschäftslogik auf Server-Ebene
 - höherer Anteil der Gesamtarbeitslast auf dem Server
 - bessere Skalierbarkeit
 - geringere Netzwerkbelastung, höhere Gesamtperformanz
 - Modularität durchbrochen
 - » proprietäre Sprachen auf Server-Ebene

Drei-Ebenen-Architekturen

- Schichten werden einzeln auf Ebenen abgebildet
 - Trennung nicht unbedingt physikalisch

- Höhere Flexibilität
- Bessere Performanz
- Größere Sicherheit

- Höhere Komplexität
- Höheres Kommunikationsaufkommen



Komponenten-Architekturen für die Server-Ebene

- Mittlere Ebene entscheidend
- Komponenten / Application Server
 - ursprünglich kein einheitlicher Standard
 - Idealziel: offener Markt

- Microsoft Distributed interNet Applications Architecture
 - Produktlösung
 - herstellerübergreifende Kompatibilität eingeschränkt

- OMG Common Object Request Broker Architecture
 - offene Spezifikation
 - beschreibt verteilte Objekte und ihre Interaktion

Sun Microsystems Java 2, Enterprise Edition

- Zunächst Entwicklung von Einzeltechnologien für die Nutzung in Unternehmen
- Schlechte Abstimmung, Unklarheiten, Widersprüche

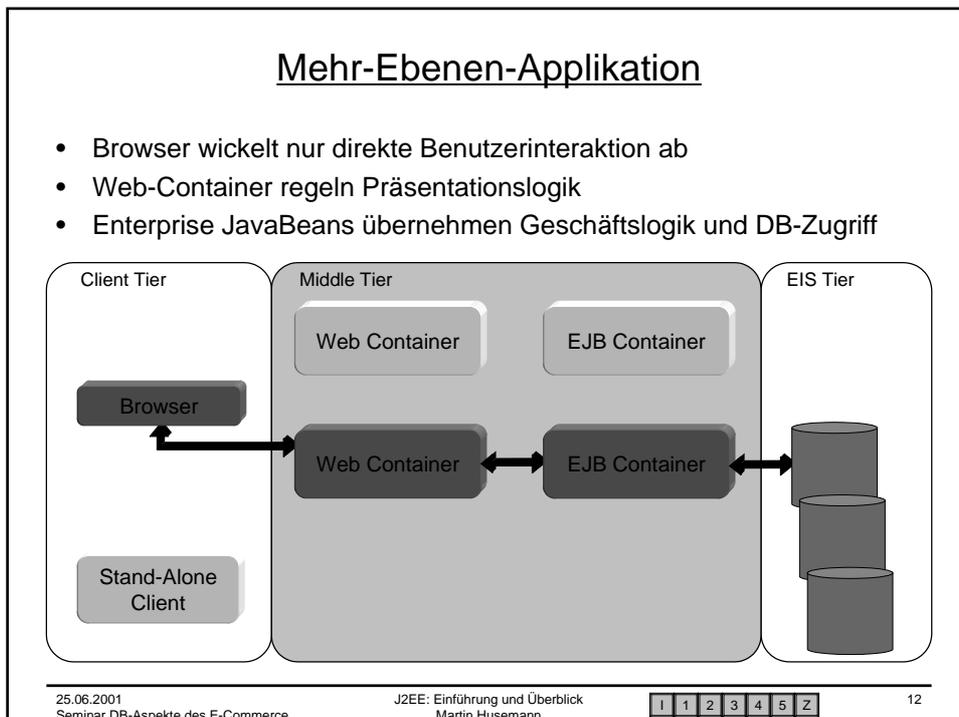
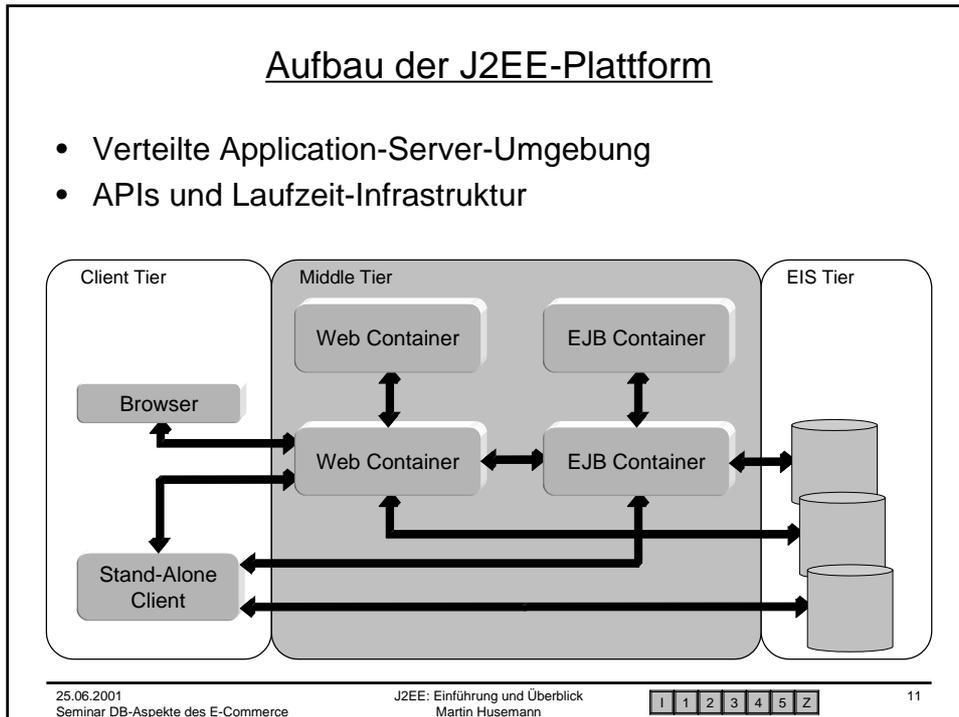
- Java 2: Micro, Standard, Enterprise Edition
- Integrierte Plattform für die Entwicklung portabler Software-Komponenten auf Server-Ebene
- Spezifikation, nicht Produkt (analog zu CORBA)

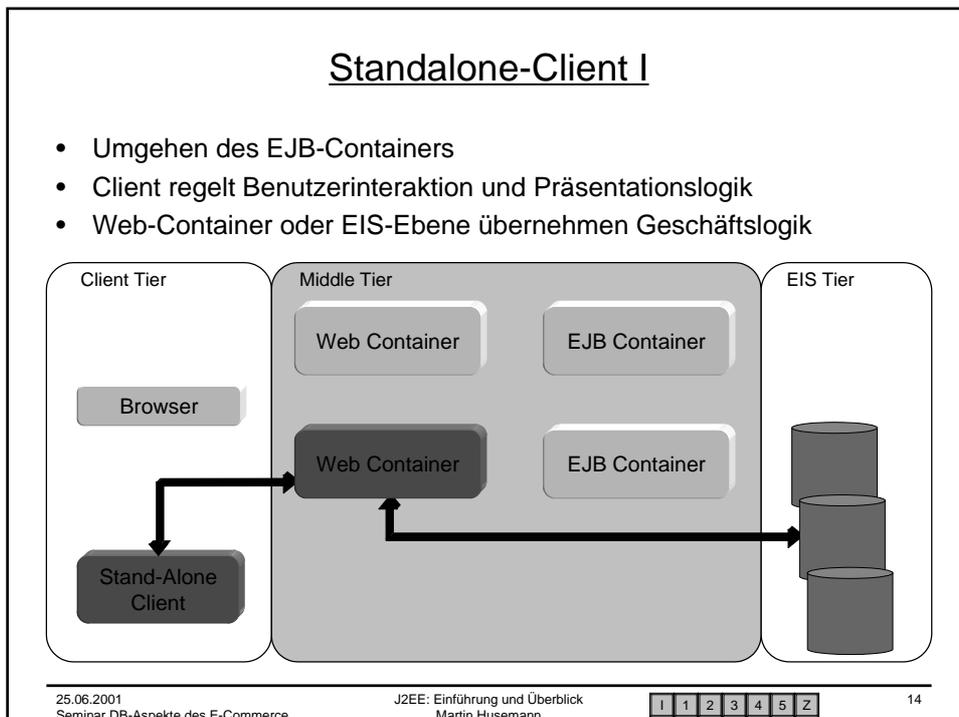
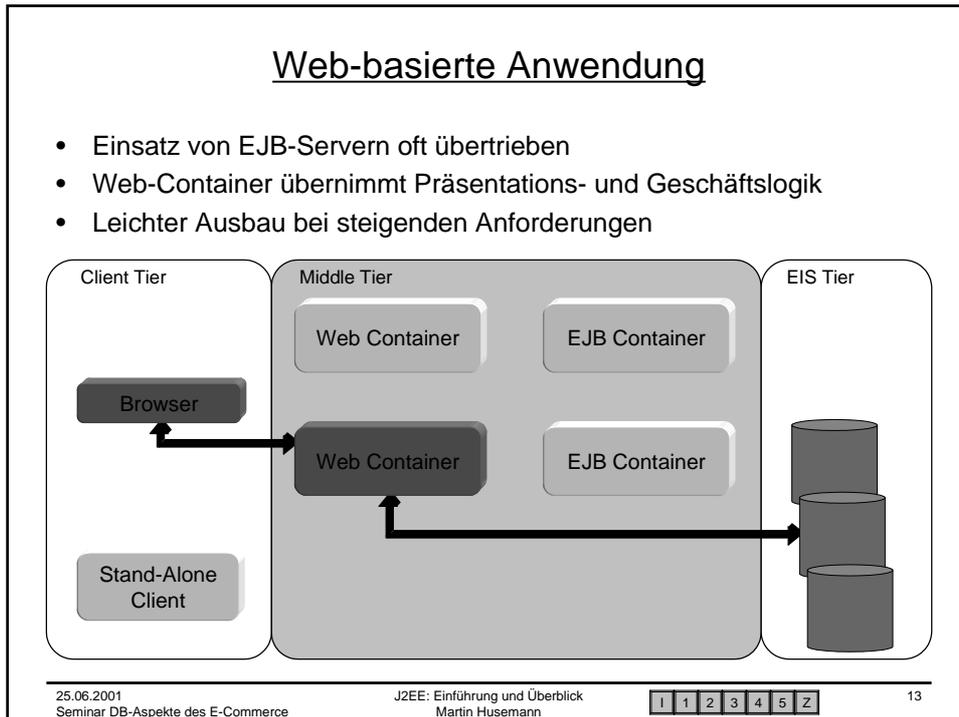
- Gegenüber CORBA zusätzlich Festlegung von Schnittstellen und Verhalten auf Komponentenebene

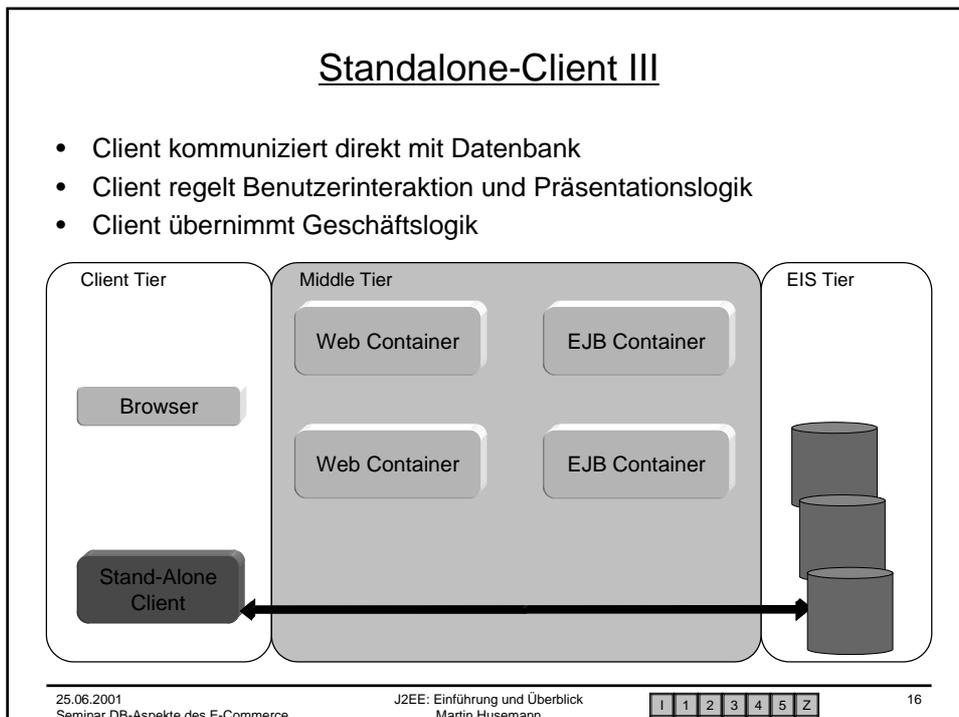
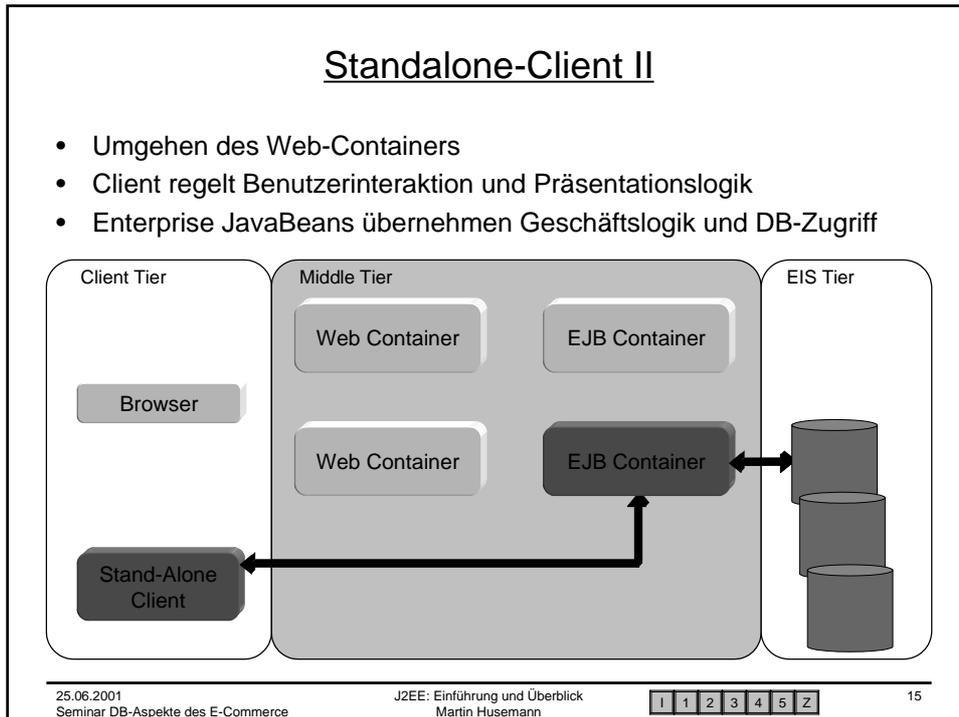
Aufbau der J2EE-Plattform

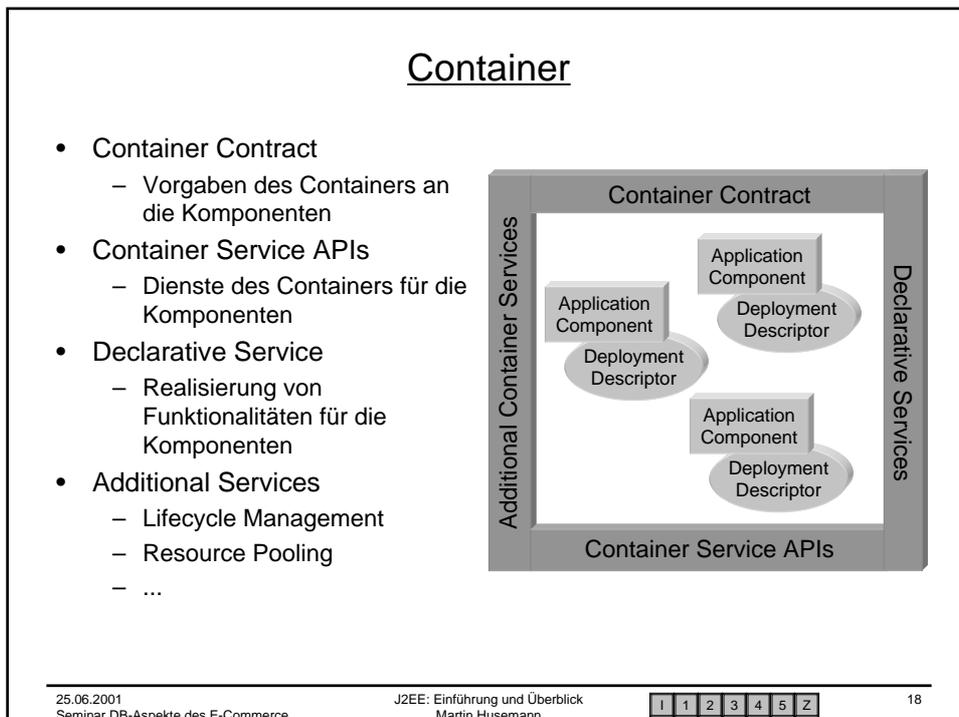
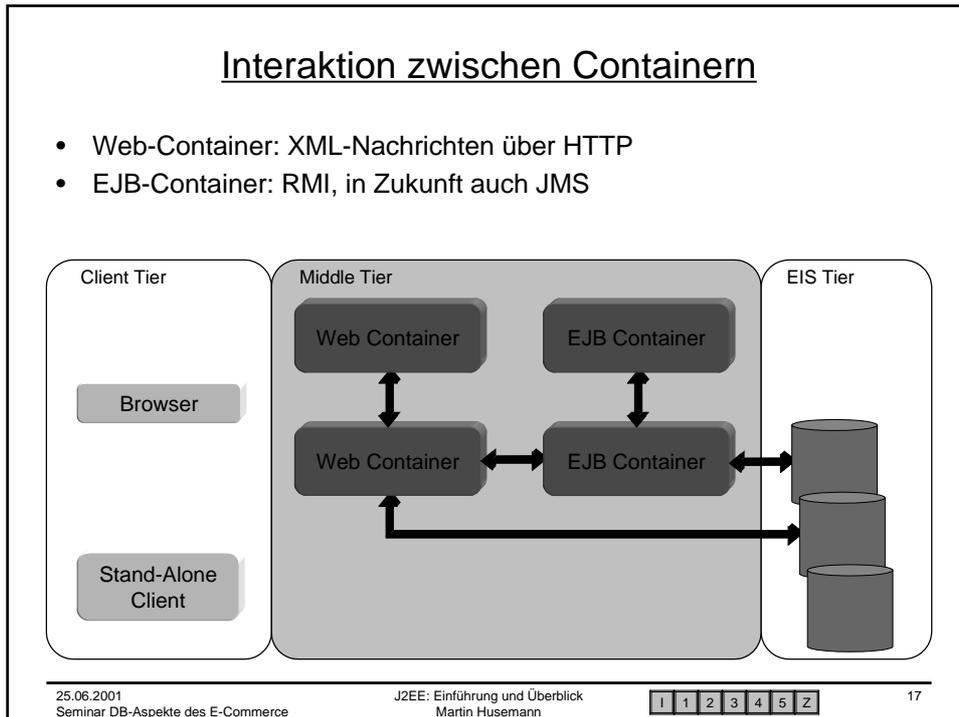
- Verteilte Application-Server-Umgebung
- APIs und Laufzeit-Infrastruktur

- Client Tier
 - Browser, Java-Applications
- Middle Tier
 - Geschäftslogik in Enterprise JavaBeans
 - Container als Laufzeitumgebung
- EIS Tier
 - Zugriff auf Datenbanken über einheitliches API



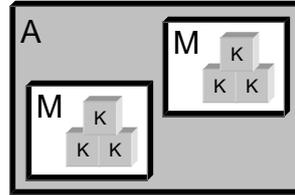






Technologien und APIs (I)

- **Komponenten-Technologien**
 - Komponente → Modul → Applikation
- **Java Server Pages und Servlets**
 - Verarbeitung von HTTP-Requests
 - JSP: um dynamischen Inhalt ergänzte Webseite
 - Servlet: aktive Komponente zum Ausbau eines Webserver
- **Enterprise JavaBeans**
 - Geschäftslogikkomponenten auf Server-Ebene
 - Session Beans / Entity Beans



Technologien und APIs (II)

- **Service-Technologien**
- **Java DataBase Connectivity (JDBC)**
 - Anbindung an relationale Datenbanken
- **Java Transaction API (JTA)**
 - implementierungsunabhängige Unterstützung von Transaktionen
- **Java Naming and Directory Interface (JNDI)**
 - Namen- und Verzeichnisdienst
 - wird von Enterprise JavaBeans genutzt

Technologien und APIs (III)

- Kommunikations-Technologien
- Remote Method Invocation (RMI)
 - erlaubt Nutzung von Methoden entfernter Objekte
 - Aufruf über lokal definierte Interfaces
- Java Message Service (JMS)
 - API für die Nutzung von Nachrichten-orientierter Middleware
 - asynchroner Nachrichtenaustausch
- JavaMail
 - Zugang zu den wichtigen Protokollen für Email-Nutzung

Zusammenfassung

- Mehrschichtige Systemarchitekturen unumgänglich
- Drei-Ebenen-Architekturen von Vorteil
- J2EE: Framework ohne Bindung an konkrete Produkte
 - Drei-Ebenen-Architektur
 - verschiedene mögliche Systemarchitekturen
 - Container-Konzept
 - Technologien und APIs