

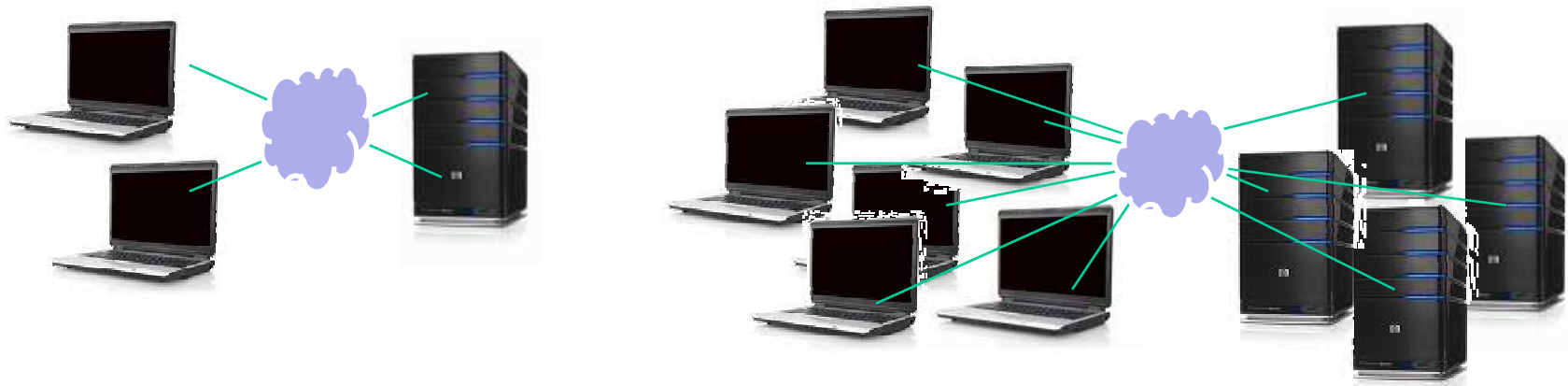
Architektur und Skalierbarkeit

Skalierbarkeit

- Skalierbarkeit
- bedeutet die Anpassung einer Software an **wachsende Last**:
 - Interaktionsfrequenz
 - Nutzerzahl
- Anpassung durch **Hinzufügen von Ressourcen**.
- **Ressourcenbedarf** zur Erhaltung der Servicequalität
 - gute Skalierbarkeit:
lineare Abhängigkeit zwischen Last und Ressourcenbedarf
mit geringem Faktor
 - z.B. zehnfache Last -- doppelter Ressourcenbedarf
zwanzigfache Last -- vierfacher Ressourcenbedarf

Skalierung

- Skalierung erfordert Hinzufügen von Ressourcen
 - Rechner (meist Server), Speicher, Datenbanken
 - statisch:
Feste Vergrößerung der Konfiguration zur Aufnahme höherer Lasten
 - dynamisch:
Automatische Anpassung der Konfiguration an die augenblickliche Last



Beispiel: Einfache statische Skalierung

Webshop

- anfangs ein Server
- bei einsetzendem Erfolg 3 dedizierte Server
 - Web- und Mailserver
 - Applikationsserver
 - Datenbankserver
- später weiterer Ausbau
 - Duplizierter Applikationssserver
 - Redundanter Datenbankserver

Beispiel: Einfache dynamische Skalierung

- Bei weiter wachsendem Erfolg:
- Hosting auf einer Serverfarm
 - Lastabhängig werden Server hinzugefügt oder wieder entnommen (heute zumeist virtuelle Server)
- Architektur-Voraussetzungen:
 - klare Schnittstellen
 - **lose Kopplung** – Laufzeit-konfigurierbar
 - **Laufzeitumgebung**, die die Skalierung unterstützt

Skalierbarkeit und Architektur

- Komponentenmodell
 - Gliederung der Software in **Komponenten**
 - "Genormte" Schnittstellenspezifikation
 - "Genormte" **Interaktion und Kollaboration**
 - Infrastruktur bietet
 - **Verteilung und Bindung (z.B. Namensdienst!)**
 - Persistenz
 - Nachrichtenaustausch
 - Sicherheit, Versionierung, weitere Dienste

Komponententechnologie

- **Komponentenmodelle umfassen**
 - Spezifikation der Komponenten
 - Spezifikation der Dienste
 - Spezifikation der Laufzeitumgebung / des Servers
- **Beispiele**
 - JEE, .Net, (Corba)
- **Skalierbarkeit:**
 - Duplizierung von Komponenten und Servern beliebig möglich.
 - **Frage der Server-Einstellung – nicht der Anwendungssoftware!**

Andere skalierbare Architekturen

... ehe wir noch genauer in das JEE-Modell sehen:

- Skalierbarkeit gibt es auch in anderen Architekturen!
 - Skalierung heißt **lastabhängiges Hinzufügen von Ressourcen** mit möglichst geringem Änderungsaufwand der Software.

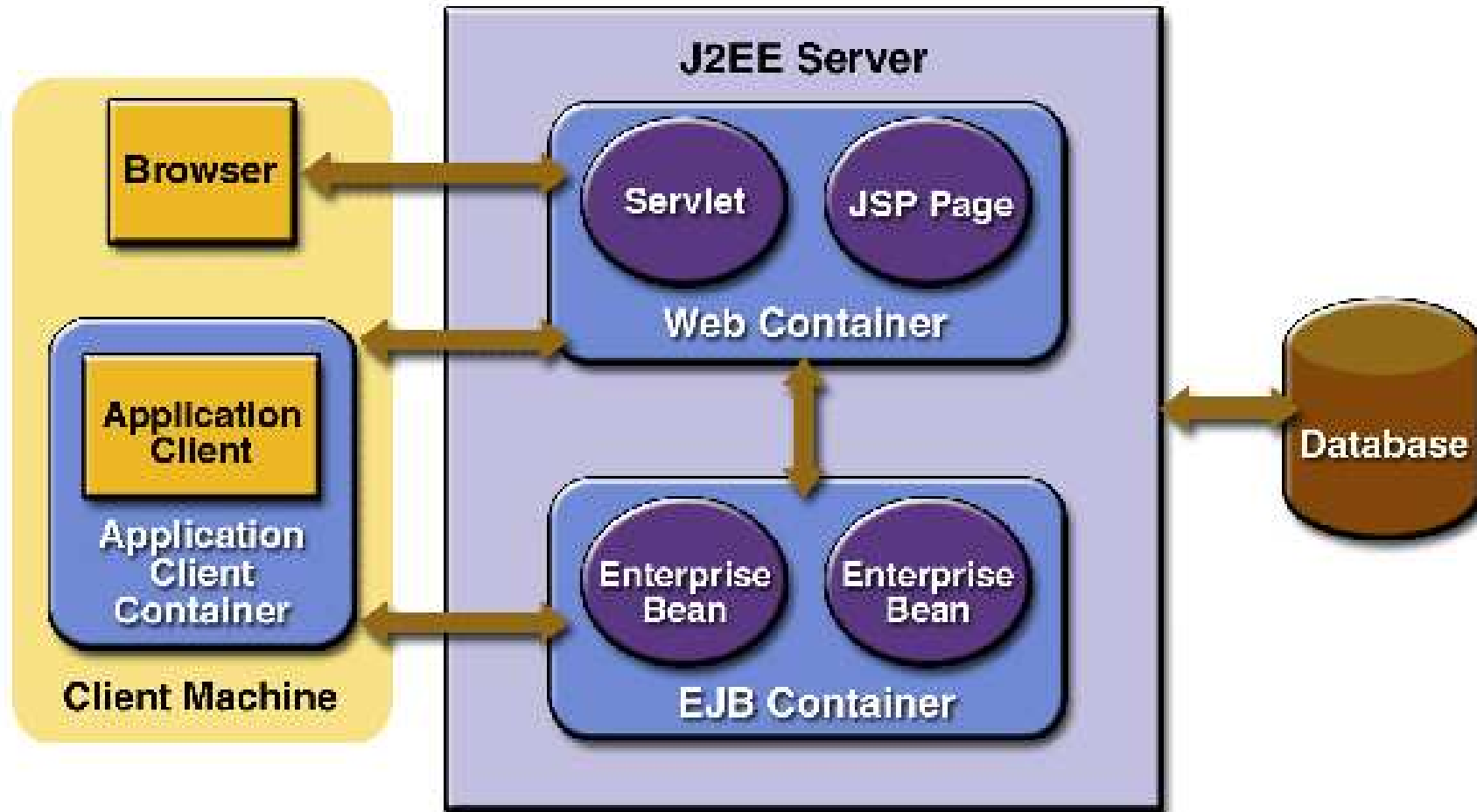
- **Peer2Peer ist der "Champion"**:
Jeder zusätzliche Teilnehmer bringt automatisch neue Ressourcen ein.



- Nicht vergessen:
 - **Gute** Skalierbarkeit bedeutet **mäßigen Ressourcenzuwachs** pro Lastzuwachs

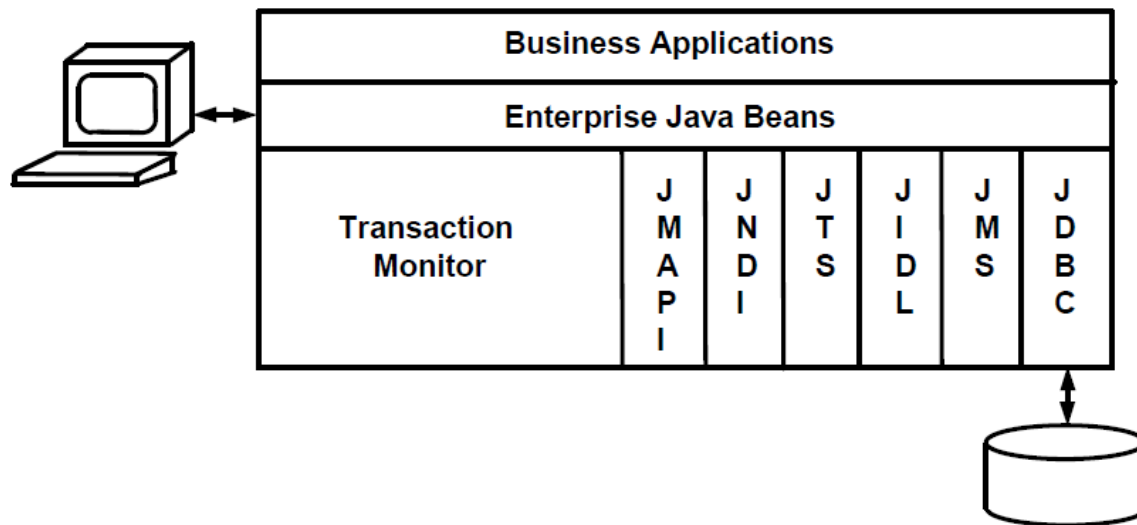
- Enterprise Java Beans (EJBs) sind die Komponenten
 - seit EJB 3 "POJOs" – keine Erweiterung bestimmter Klassen erforderlich
 - SessionBeans – Geschäftslogik
 - EntityBeans – Modell
 - MessageDrivenBeans – asynchroner Aufruf
- Laufzeitumgebung EJB-Container
 - oder "Java Application Server"
 - Instanziierung, Verwaltung und Persistenz für EJBs
 - zusätzliche "EJB-Services" wie Transaktionen, Namensdienst etc.
 - Konfigurationsmöglichkeiten für Skalierung und Load Balancing

JEE-Referenz-Architektur



EJB-Services

- JMAPI Java Management API
- JNDI Java Name Directory Interface
- JTS Java Transaction Service
- JIDL Java Interface Definition Language
- JMS Java Message Service
- JRA Java Resource Adapter API
- JSA Java Security API



Maßnahmen für die Skalierbarkeit

- Programmieren gegen **Interfaces**
 - Interfaces reduzieren die Abhängigkeiten
 - Erleichtern die Einführung performanterer Lösungen
(Nur die betroffenen EJBs müssen neu deployed werden)
- Lose Kopplung durch **Dependency Injection**
 - Reduziert die Kopplung
 - stellt dennoch eine feste Beziehung zwischen zwei Objekten her
- Extra-loose Kopplung durch **Name-Directory-Lookup**
 - Bei Bedarf wird ein Service über die Registry gefunden
 - Service-Locator-Pattern

→ *Alle drei Techniken finden Sie im 2. CRUD-Modell*

Skalierbarkeit ist nicht alles...

- Skalierbarkeit ist nur eine der nichtfunktionalen Anforderungen.
 - Umsetzung beispielhaft diskutiert
- Performance
- Sicherheit
- Verfügbarkeit
- ...
- *Praktisch jede Anforderung muss bereits im Architekturentwurf berücksichtigt werden!*

Das war's an Modellarbeit



... und für dieses Jahr!

**Frohe Weihnachten, guten Rutsch
und auf Wiedersehen 2011!**