



BEA WebLogic Server :
Der EJB 2.0 Server

Thomas Walter
Manager Enterprise Java Group
(Central / Eastern Europe)
Phone: +49-7051-77355
Mailto: thomas.walter@bea.com



Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
- Mission critical Support für EJBs
- Zusammenfassung

De-Facto Standard für J2EE Application Server

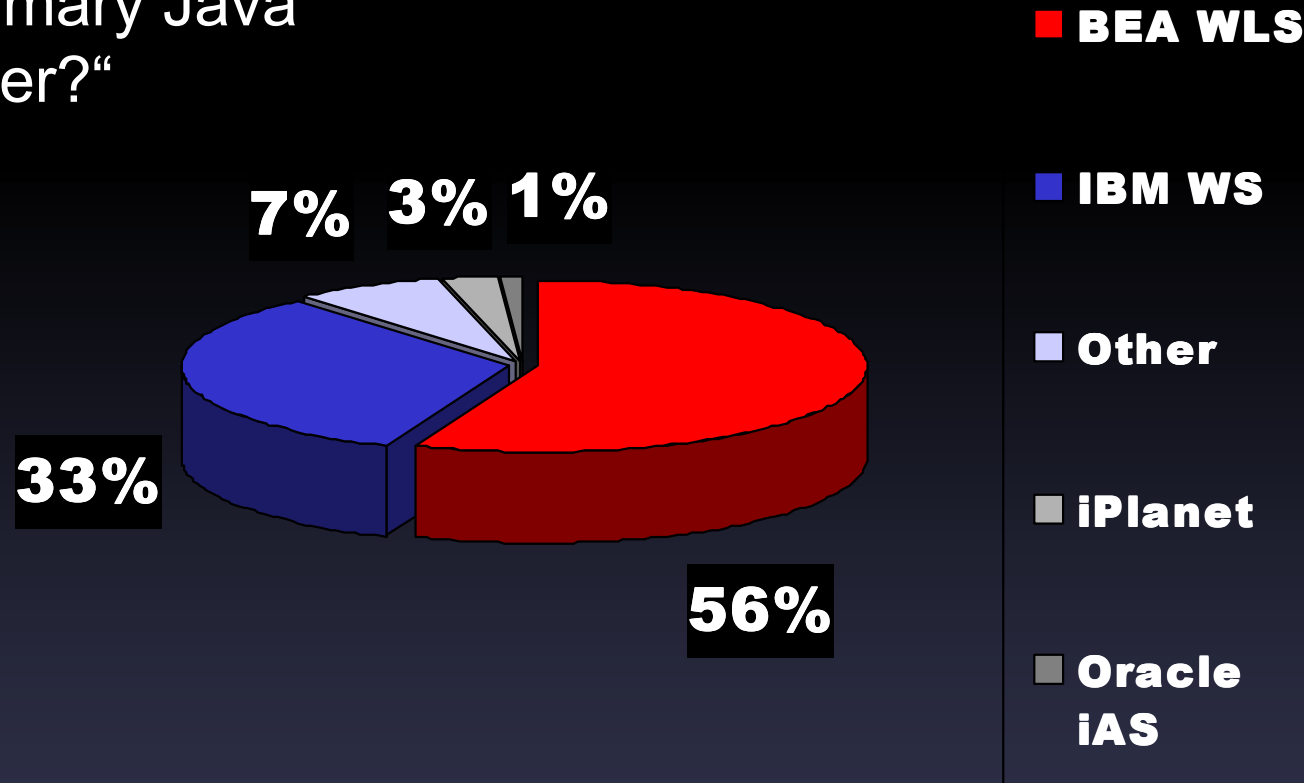


- Giga Group (M. Gilpin), June 2000: “#1 market share leader (32%), with 2 times the market share of 2nd-place IBM.”
- IDC (S. Garone), June 2000: “#1 market share in application servers and transaction servers”
- Gartner (Y. Netis), March 2000: “WebLogic Server has the largest installed base of EJB applications in the industry”

GIGA Survey 12. Dez. 2000:



„What is your primary Java Application Server?“



Quelle: "GIGA WebSurvey Results", Mike Gilpin, Dec. 2000



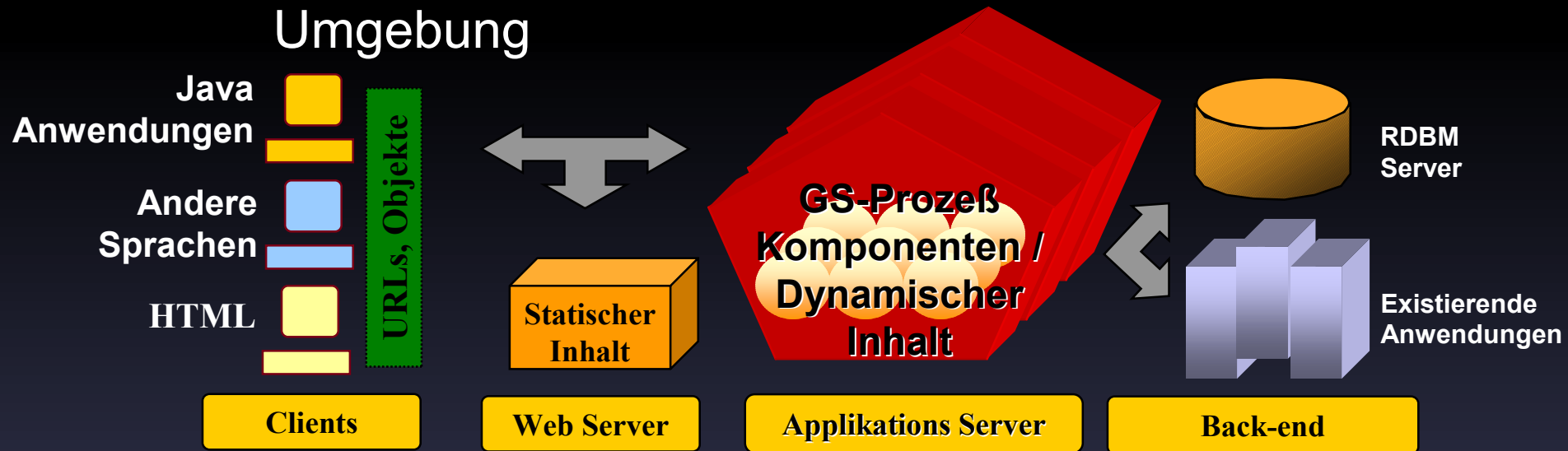
Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- WebLogic Java Application Server
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
- Mission critical Support für EJBs
- Zusammenfassung

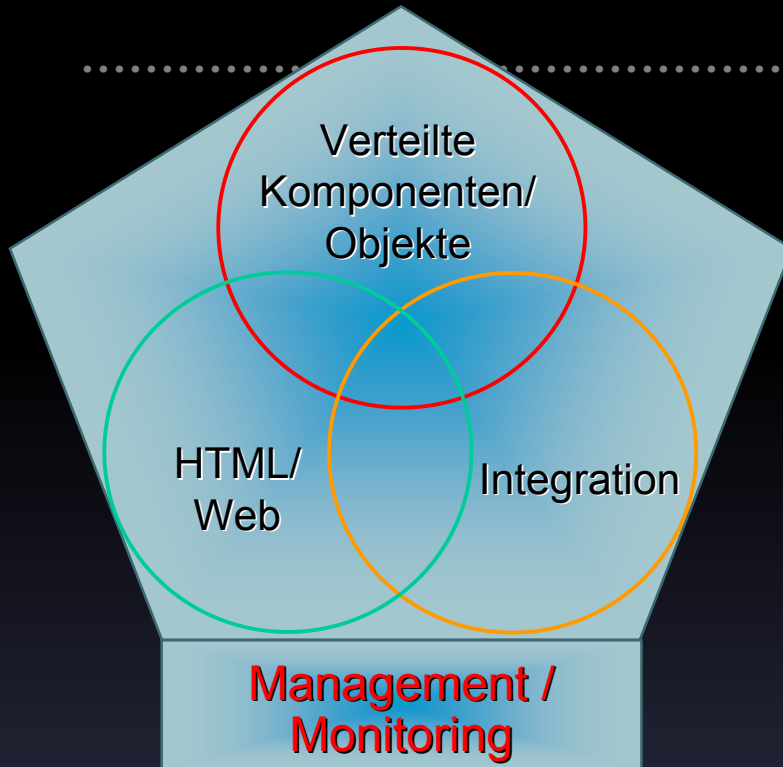


J2EE Applikations-Topologie

- Java-Applikations-Server
 - Beheimaten Anwendungen in einer “multi-tier” Umgebung



Java Chronologie: Schritt 2. Der Server

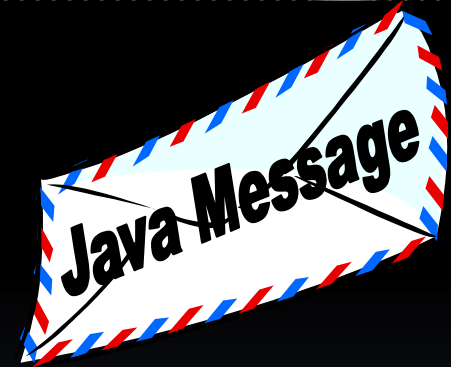


*Server basierend auf
Industrie-Standard APIs*

**J2EE 1.2
zertifiziert**

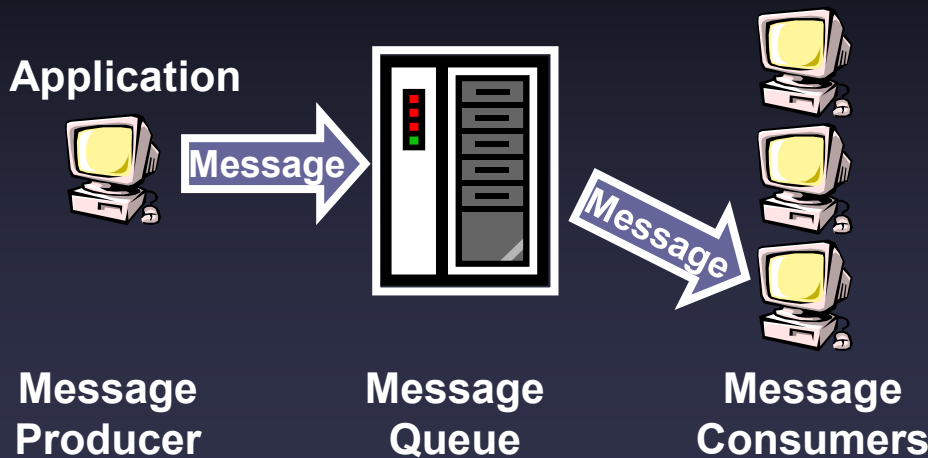
- **Verteilte Objekte / Komponenten**
 - Enterprise JavaBeans™ 2.0 (EJB)
 - Remote Method Invocation (RMI)
 - Java Transaction Service & API (JTS/JTA)
 - Java Naming & Directory Interface (JNDI)
 - Java Messaging Service (JMS)
 - Java Interface Definition Language (JIDL)
- **Web/HTML**
 - Servlets
 - Servlet session management
 - Java Server Pages (JSP) / Java HTML
- **Integration**
 - Java Database Connection (JDBC)
 - Multi-tier JDBC
 - Java Connector Architecture (JCA)
- **Management / Monitoring**
 - Java Management Extension (JMX)

WebLogic Java Message Service Der Unternehmens Message Bus

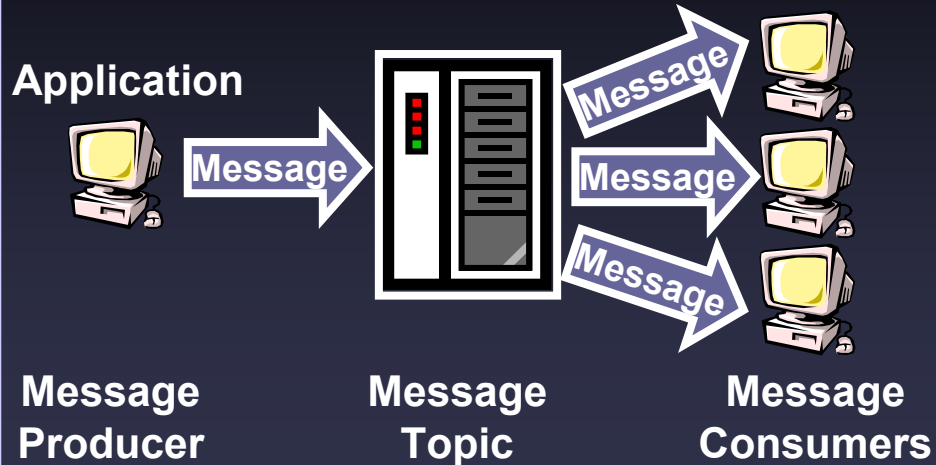


- Standard JavaSoft API (Version 1.0.1) für
 - Publish / Subscribe Model (Pub/Sub)
 - Durable möglich
 - Point-To-Point Model (PTP)
- Optionale Server Session Pools (serverseitiges parallele Verarbeitung von Messages)

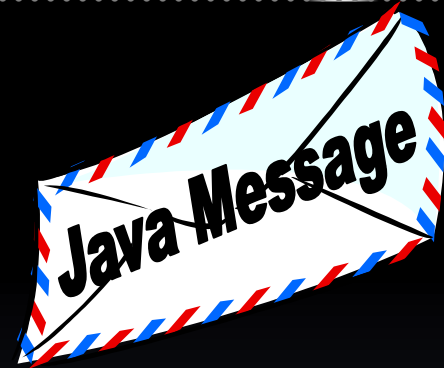
PTP - Modell



Pub / Sub - Modell

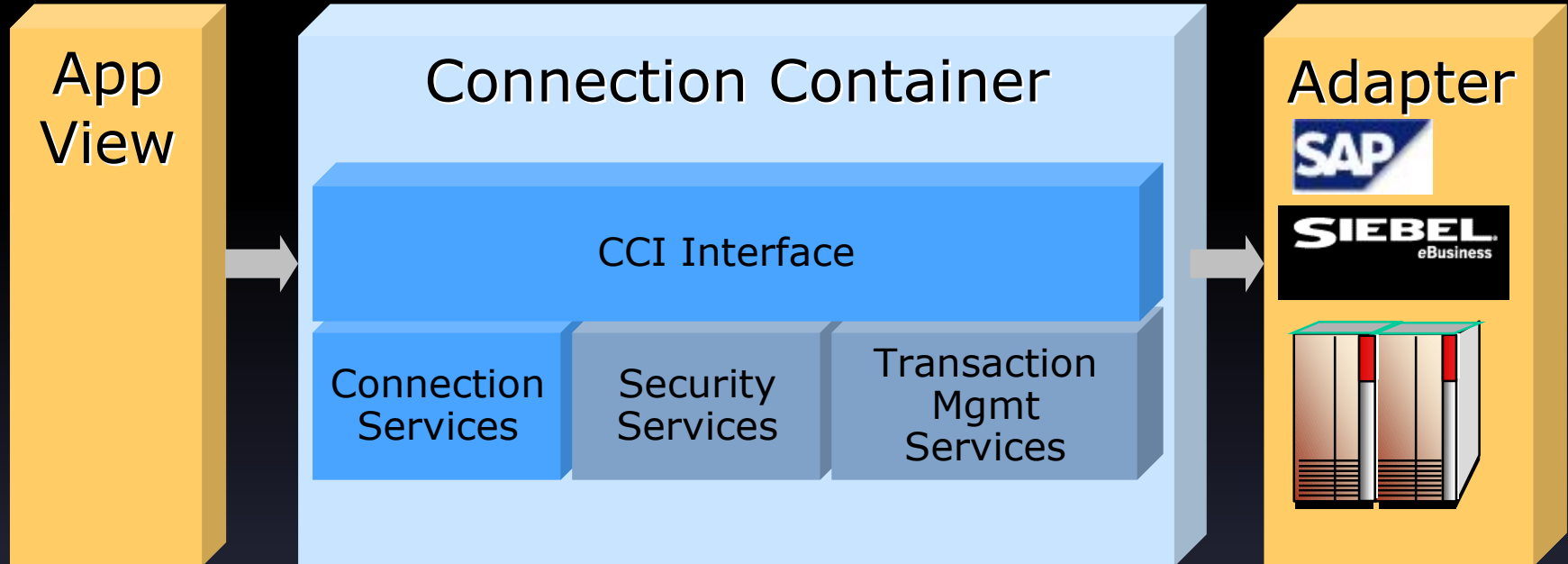


WebLogic Java Message Service Quality of Service



- Messages können sein
 - NON-PERSISTENT oder
 - PERSISTENT (DBMS, File)
- JMS Operationen können Teil einer Transaktion sein
 - *Transacted* JMS Sessions zum Mischen von EJB und JMS (optional in Spezifikation)
 - *Non-transacted* JMS Sessions benutzen JMS Acknowledgement
- BEA High-Performance Delivery
 - Point-to-Point TCP/IP oder Multicast

Das neue EAI Paradigma: Java Connector Architecture

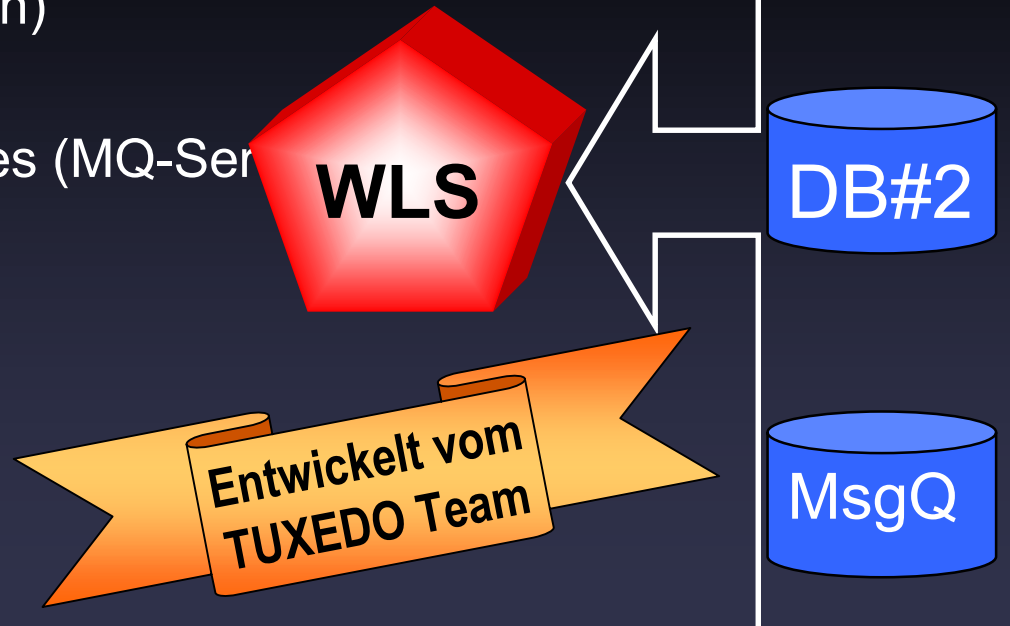


- Teil des J2EE Industriestandards
- Unterstützt Pooling, Security und Transaction Management
- BEA Adapter erhältlich für
 - SAP R/3, Mainframe (CICS / IMS) & Siebel in Vorbereitung
 - Mailadapter & DBMS Adapter als Quelltext

Garantie für Konsistenz: Distributed Transaction Manager



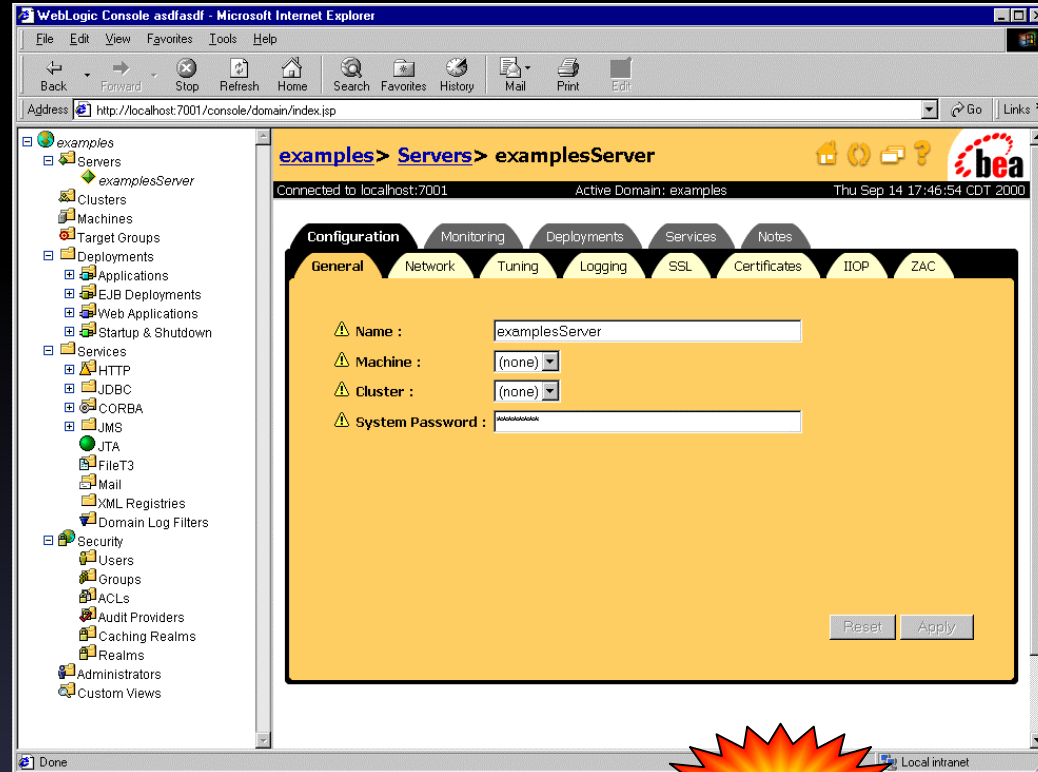
- Komplette in Java
- 2PC mit jedem Storage Provider, der XA-Resource Interface unterstützt, z.B.:
 - DBMS via JDBC 2.0 XA Treiber (Oracle OOTB)
 - JMS (alle Persistenzarten)
 - EJBs
 - Externe Message Queues (MQ-Server)
 - JCA Adapter
- Transaktionsinfektion
 - über mehrere Server
 - mit & ohne Cluster



Management und Monitoring: Basierend auf JMX Standard



- Zentrale Überwachung & Administration
- Browserbasiert
- Hot-Deploy von EJBs / Servlets
- Java Management eXtensions
 - Teil von J2EE 1.3
 - Tivoli, BMC Patrol, ...
 - Zusätzlich SNMP Agent



Neu!

WebLogic Server XML Support als Evolution



- **Welle 1: Parser & Übersetzer**
 - SAX, DOM, JAXP
 - Parser Generator
 - XSLT
- **Welle 2: Objekte**
 - XML Schema Repository
 - XML Data Binding Reference Implementation
- **Welle 3: WebServices**
 - Simple Object Access Protocol (SOAP)
 - Web Services Definition Language (WSDL)
 - Universal Description, Discovery and Integration (UDDI)





Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
- Mission critical Support für EJBs
- Zusammenfassung

BEAs EJB Führerschaft zu 2.0: Ihr Time-to-Market



- **Führender Enterprise JavaBeans Anbieter**
 - EJB Technologie seit Sept. 97 in Auslieferung
 - EJB 1.0 April 98
 - EJB 1.0 Produktion Juli 98
 - Clustered EJB Nov. 98
 - EJB 1.1 Mai 1999
 - EJB 2.0 Early Adopters Juni 2000
 - EJB 2.0 Generelle Verfügbarkeit Dez. 2000
- **Ausgereiftester skalierbarster EJB
Application Server auf dem Markt**

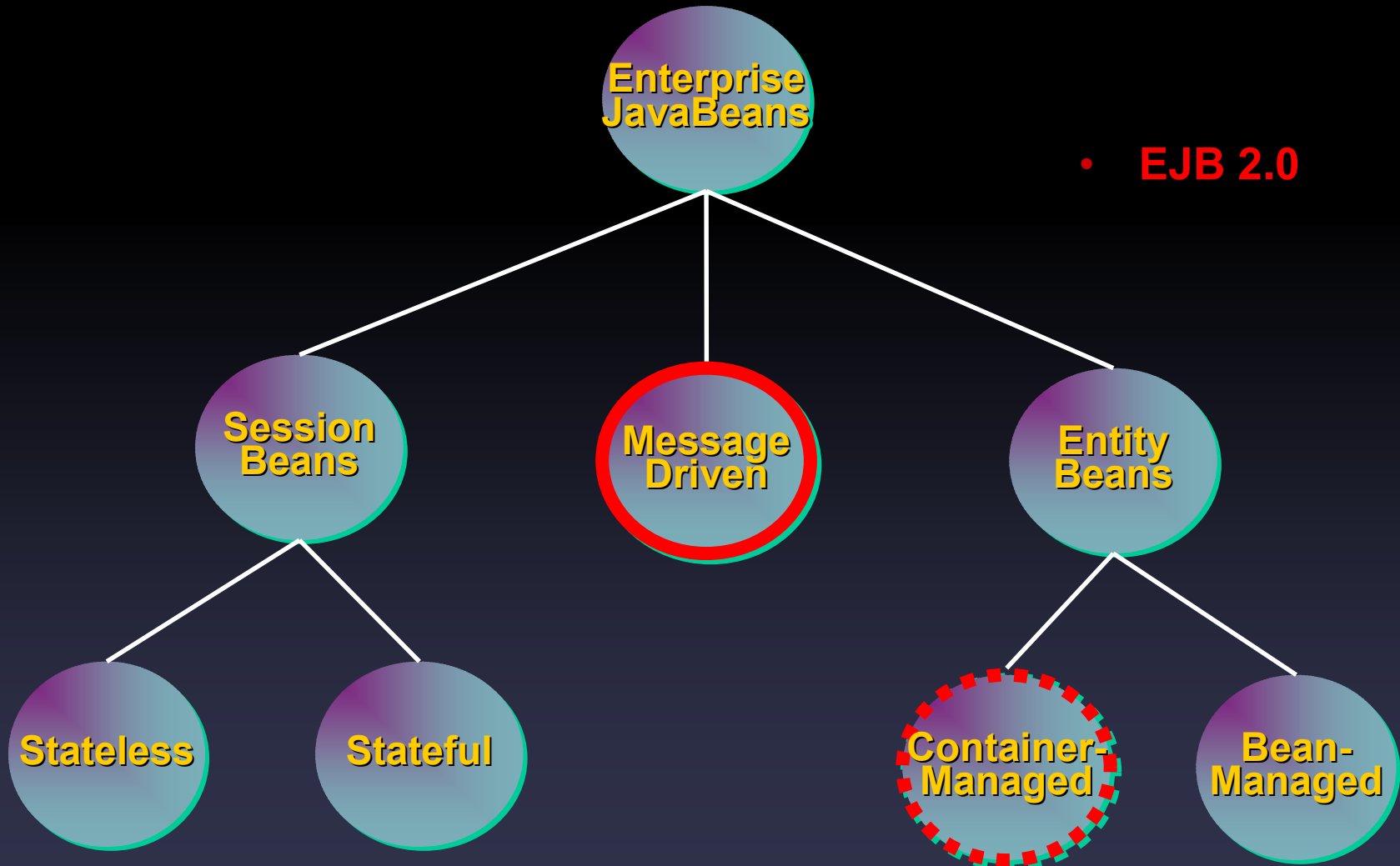


EJBs: Effektivität und Konsistenz

- Entwickler bekommen
 - Persistenz
 - Relationen, Abstraktion von DBMS, Queries, Lazy Loading
 - Transaktionen
 - Konsistenz in der Applikation
 - Security
 - ACLs und Laufzeitüberprüfung
 - Naming
 - Lokationstransparenz über Clients und Server hinweg
 - Lifecycle und Caching
 - “Swapping” auf Applikationsebene
 - Concurrency-Modell
 - Keine Low Level Multithred Programmierung
- Kurzum Mannjahre gespart.....



EJBs: Eine hochspezialisierte Familie



• EJB 2.0



EJB 2.0 Home-Methods

- Dürfen nicht beginnen mit
 - create, find, remove
- Können alles zurückgeben
- Sind in etwa wie statische Methoden
 - Beziehen sich nicht auf eine Instanz
 - Sollten sich auf EJB-Klasse beziehen
- Bsp.: `public int ageOfYoungestEmployee`
 - Macht evtl. `select min age of emp_table`



EJB 2.0 Message Driven Beans

- Asynchron
 - Container startet aufgrund JMS Message
 - Queues oder Topics
- Unsichtbar für Client
 - Kein Remote Interface
 - Kein Client conversational State
- Implementieren JMS MessageListener Interface
 - onMessage() Methode
- Sind transaktionell





Message Driven Beans: Queue Beispiel

Bei Serverstart
Queue-Aufbau u.
Subscriptions

Client



JMS-Session



MDB
Pool





Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
 - Architektur
 - Relationen
 - EJB-Query-Language
- Mission critical Support für EJBs
- Zusammenfassung

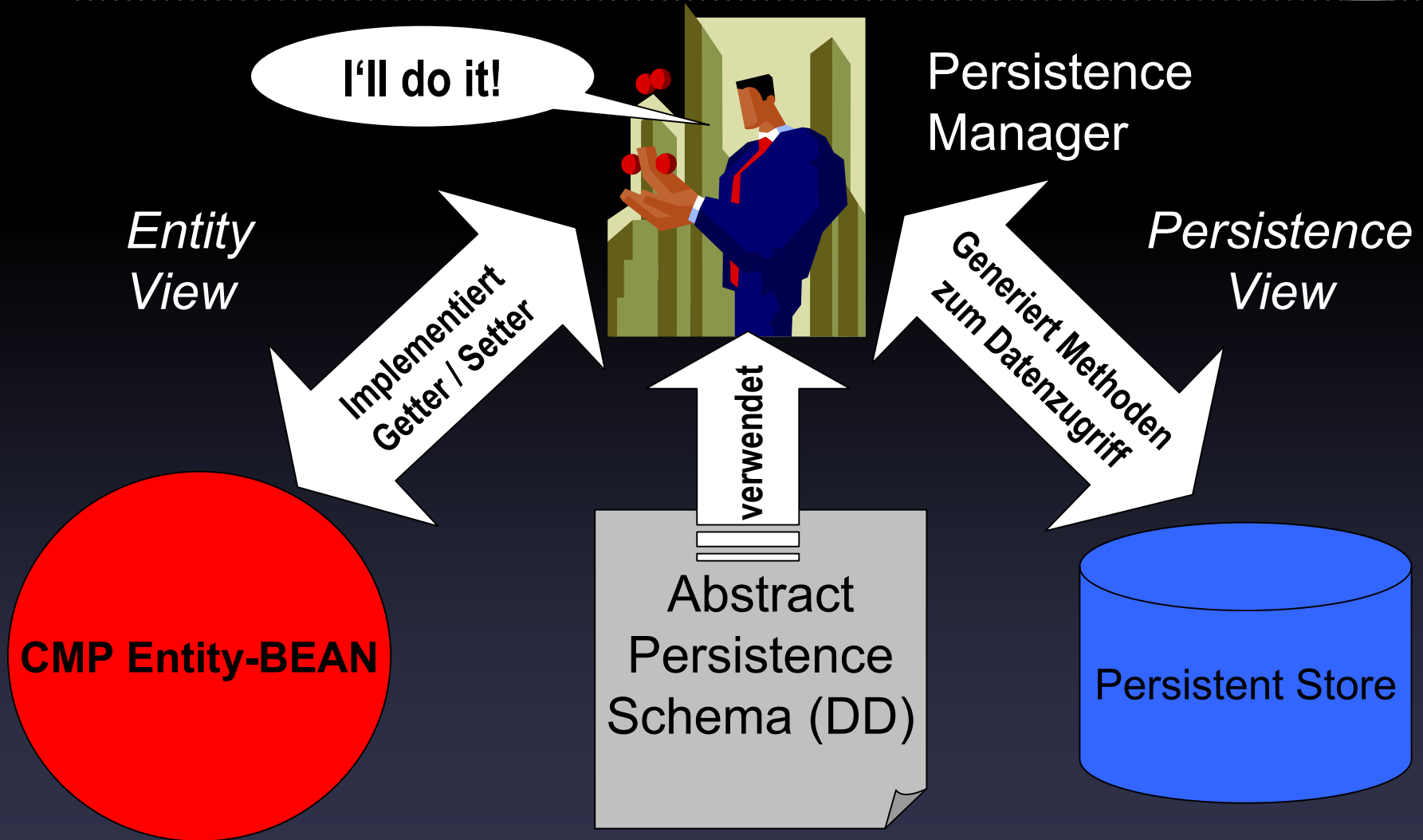
Container Managed Persistence in EJB 2.0



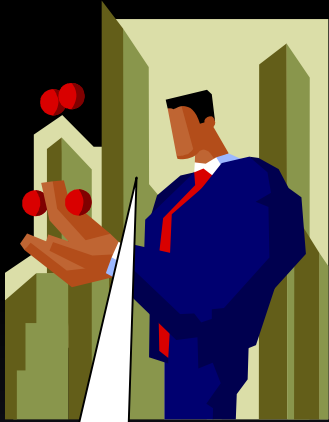
- Ziele
 - Separation von Objektdefinition und dessen persistentem Zustand
- EJB 1.1 CMP wird immer noch unterstützt
- Wie immer eine API
 - Quality of Service obliegt dem Implementierer



CMP in EJB 2.0: MVC-Pattern



Persistence Manager



I'm
always
there!

- Generiert Methoden für persistenten Datenzugriff
- Verwaltet Mapping zwischen logischen und physischen Relationen & kümmert sich um referentielle Integrität
- Ist zur Laufzeit verfügbar
- *...sorgt also für logische Unabhängigkeit des EBs von der Datenquelle*

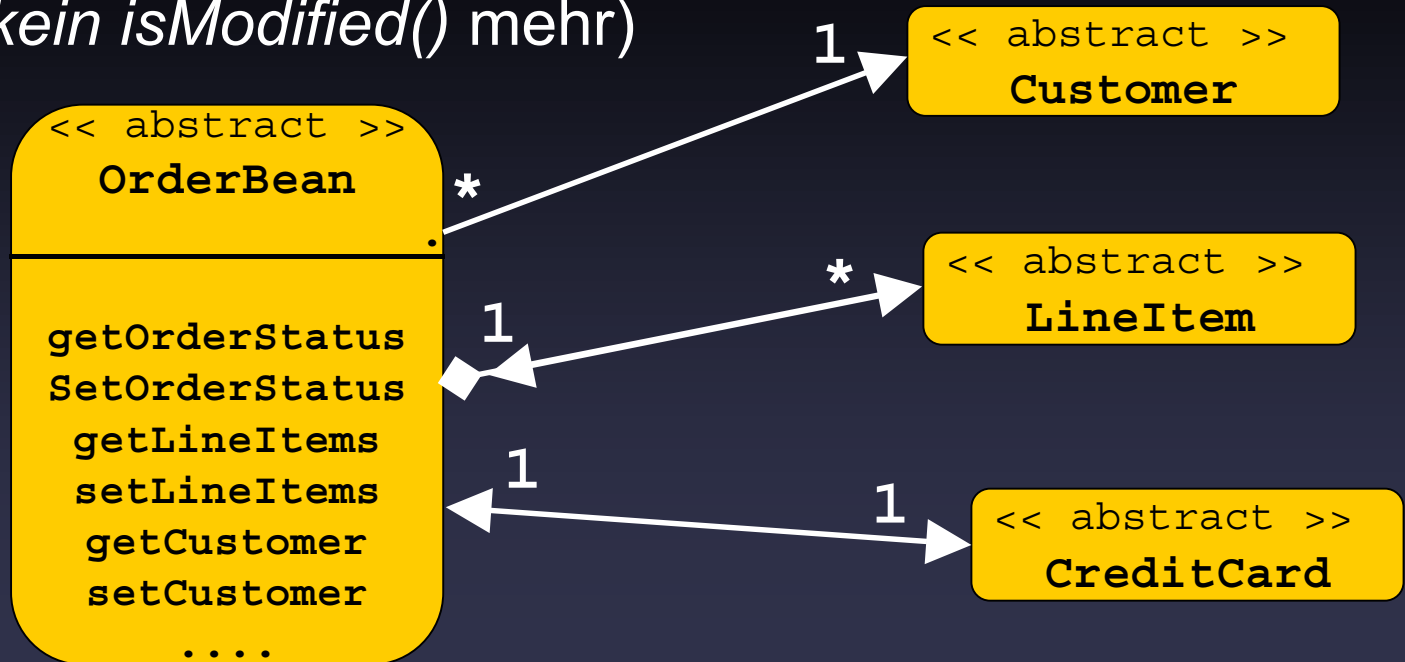
Abstract Persistence Schema

- Erstellt durch den Bean Provider als Teil des Deployment Descriptors
- Persistente Eigenschaften eines Objekts
 - Fields von Entity Beans (**CMP**-Fields)
 - Daten um Relationen zu speichern (**CMR**-Fields)



CMP und CMR Fields sind virtuell

- Zugriff auf Beans nur noch durch Accessor-Methoden
 - public, protected und abstract
 - Erlaubt Herstellern Optimierung von Updates!
(kein *isModified()* mehr)





CMP und CMR Fields sind virtuell

```
public abstract class OrderBean implements
    javax.ejb.EntityBean{
    private EntityContext context;
    ....
    // getters and setters for the cmp fields
    public abstract int getOrderStatus();
    public abstract void setOrderStatus(int orderStatus);

    // getters and setters for the relationship fields
    public abstract Collection getLineItems();
    public abstract void setLineItems(Collection
    lineitems);
    public abstract Customer getCustomer();
    public abstract void setCustomer(Customer customer);
```



Inhalt

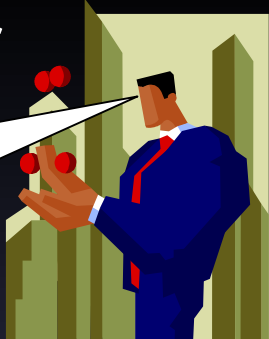
- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
 - Architektur
 - Relationen
 - EJB-Query-Language
- Mission critical Support für EJBs
- Zusammenfassung



Relationen

- Sind zusammen mit CMP-Fields Teil des abstrakten Persistence-Schemas
- Implementierung der Relationen und CMP-Felder obliegt Persistence Manager
- Persistence Schema lebt im DD
 - Standard-Deskriptor: `ejb-jar.xml`
 - WL-CMP-Deskriptor: `weblogic-cmp-rdbms-jar.xml`

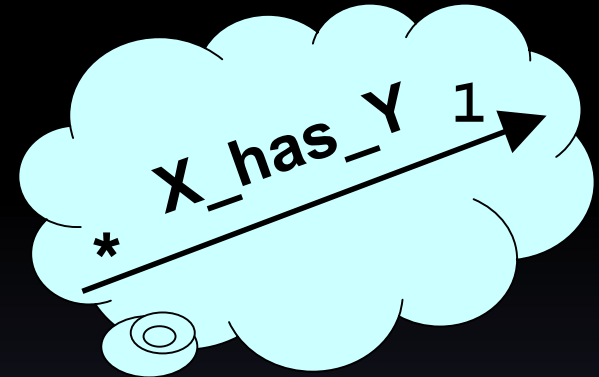
I'll implement!





Relationen-Eigenschaften

- Name
- Navigierbarkeit
 - Unidirektional / Bidirektional
- Multiplizität
 - One-to-one / One-to-Many / Many-to-Many
- Reichweite
 - Nun immer lokal (Proposed public final Draft 2)



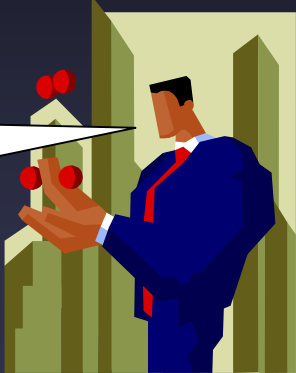
Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
 - Architektur
 - Relationen
 - EJB-Query-Language
- Mission critical Support für EJBs
- Zusammenfassung

EJB Query Language: Überblick



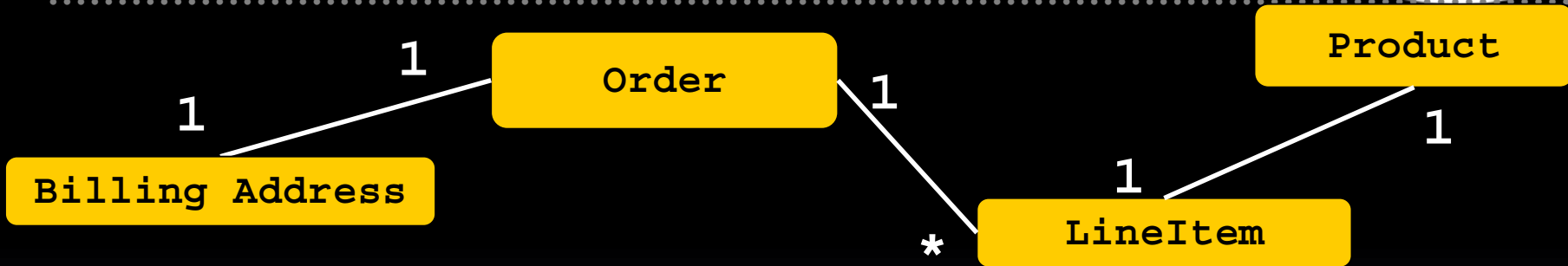
- Portabilität durch:
 - einheitliche EJB-QL Syntax
 - Queries nicht im Code, sondern im DD
 - Signaturen müssen natürlich in den Code
- Persistence Manager implementiert die Query-Methoden
- Zwei Arten von Query-Methoden
 - Finder
 - Select Methoden

A cartoon illustration of a man in a blue suit and red tie, holding three red balls in his hands. He is standing in a green, stylized architectural setting.

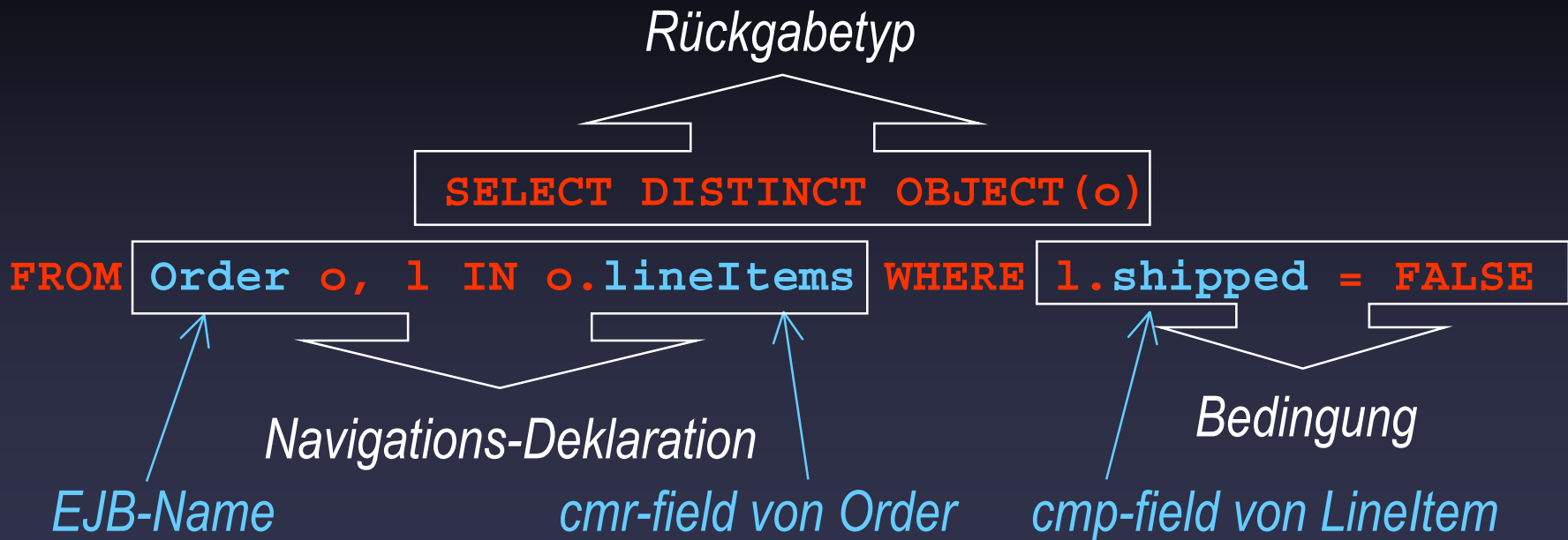
PM: I'll implement
your Methods!
Yeah!



EJB Query-Language: Beispiel



“Gebe alle Bestellungen die nur teilweise geliefert sind”





EJB Queries: Parametrisierung

- WHERE-Bedingungen können parametrisiert werden
 - ?*n* entspricht n.tem Parameter des Finders/Selects

WHERE l.product.name = ?1

- Übliche Operatoren in Bedingung erlaubt
 - AND, OR, NOT, =, >, >=, <, <=, <>, +, -, *, /
 - BETWEEN *expr1* AND *expr2*, IN (*expr1*, ..., *exprN*)
 - LIKE pattern etc.



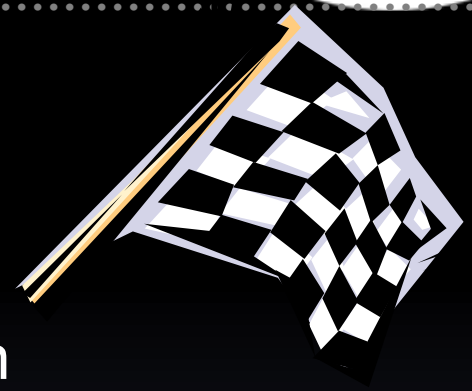


Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
- Mission critical Support für EJBs
 - Tuned Read / Writes
 - Cachingstrategien
 - In-Memory Replikation von stateful Session Beans
- Zusammenfassung

WebLogic tuned Read / Writes

- Automatische tuned Writes durch EJB 2.0 setXXX()-Methoden
- Tuned Reads über Field Groups möglich
 - cmp- und cmr-Felder
 - Feld kann zu mehreren Gruppen gehören
 - Relationen können Gruppe haben
- Bsp.: Adressen von Kunden werden nur gelesen wenn wirklich gebraucht



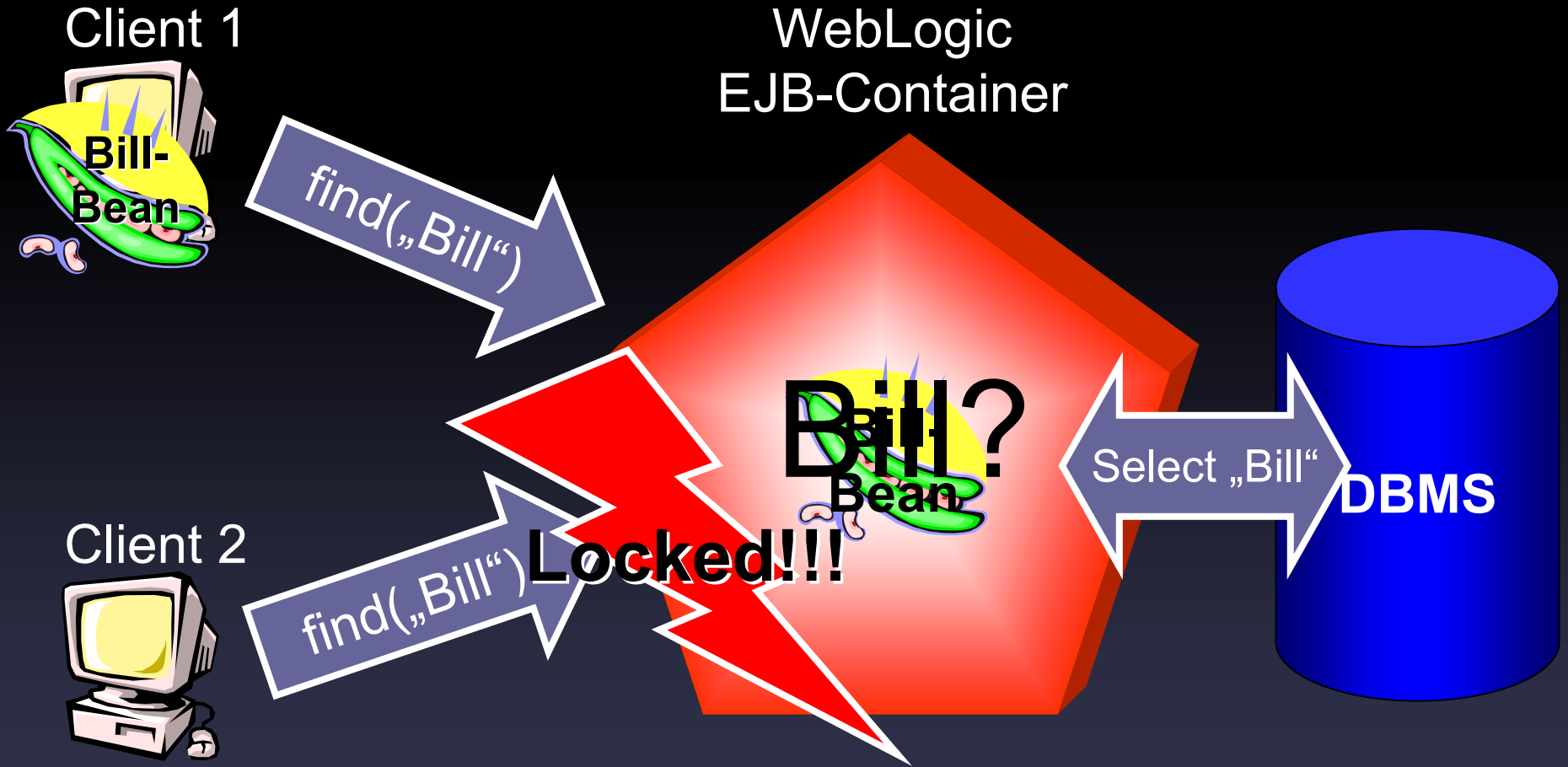
WebLogic tuned Read / Writes

- Finder
 - Laden des Beans kann verzögert werden bis zu erstem Methodenaufruf
 - Kann auch Gruppe haben (Preloading)
- Update-Verzögerung bis TX-Ende
 - Mehrere Methodenaufrufe auf Bean treffen DBMS nur einmal



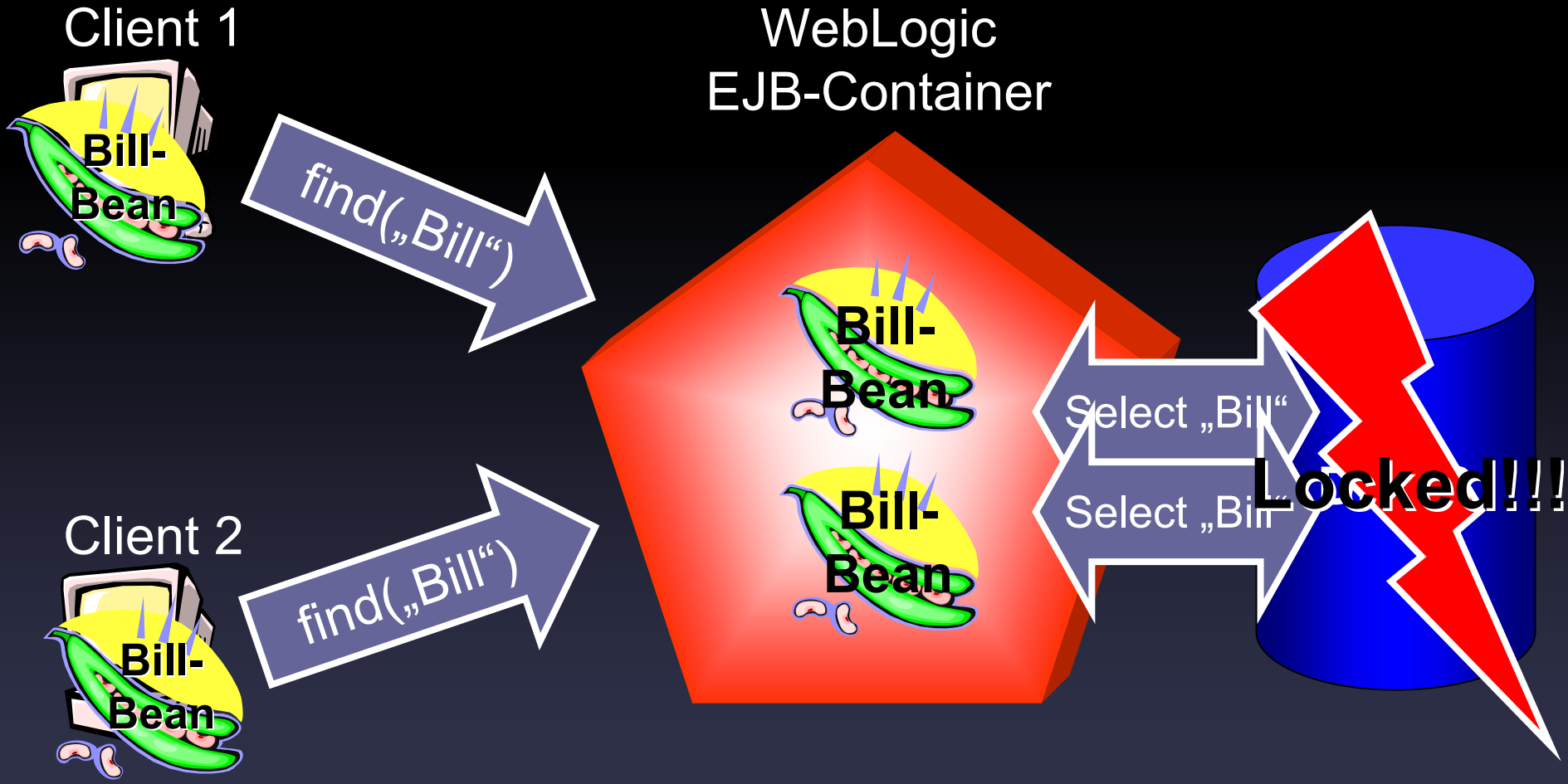


Caching Strategien: Exclusive

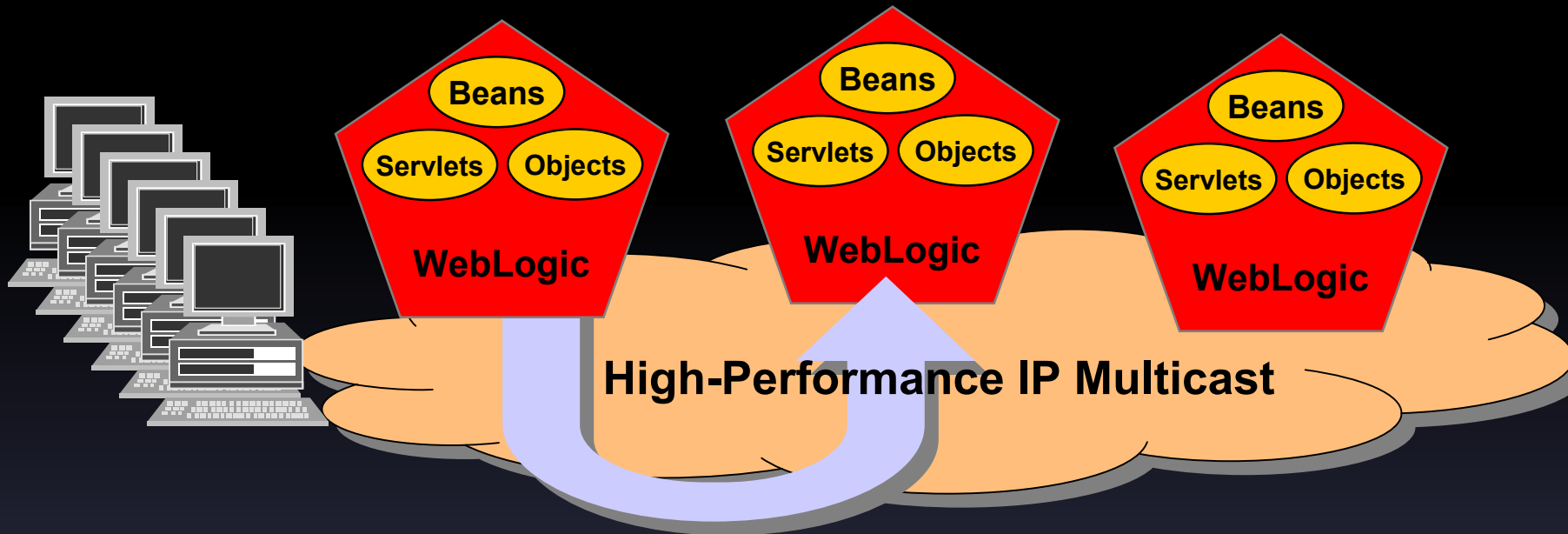




Caching Strategien: Database



Skalierbarkeit und Ausfallsicherheit: WebLogic Clustering

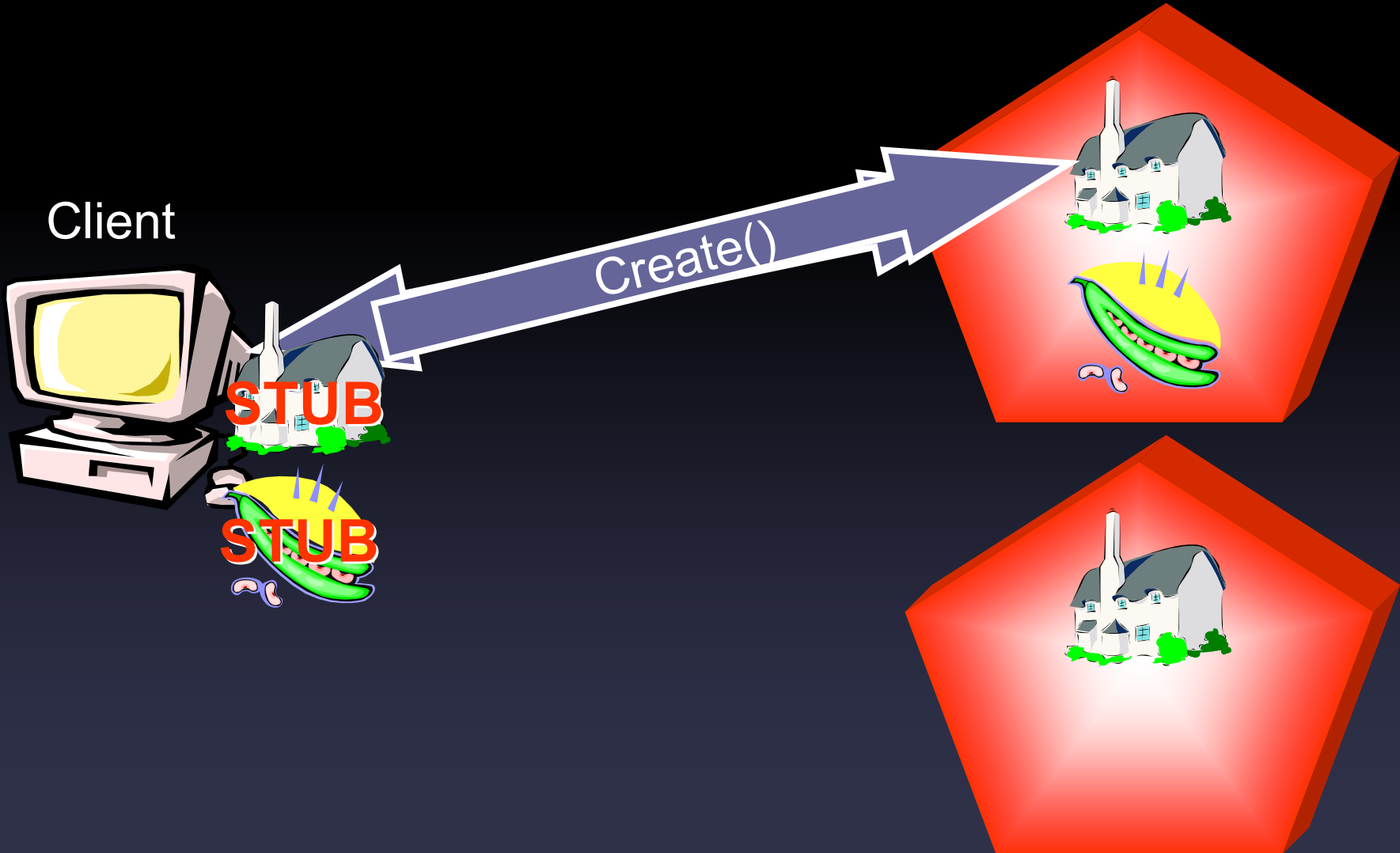


- Transparente Replikation und Failover (In-Memory)
 - Servlets, EJBs
- Automatisches Load Balancing
 - Random, Round-Robin, weight based, selbstdefiniert
- Automatisches Failover
 - Auch stateful Session Beans



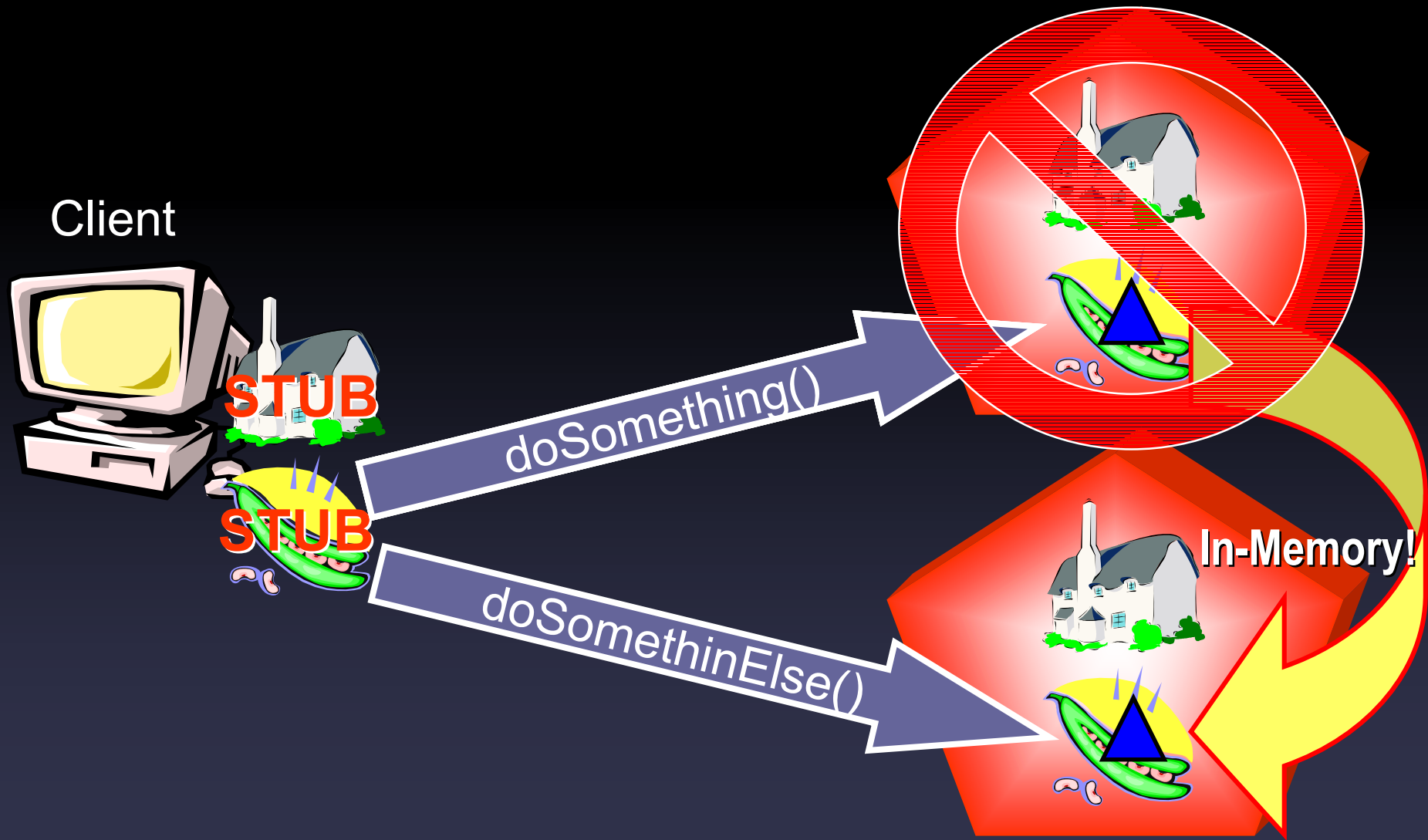


Stateful Session Beans: In-Memory Replication





Stateful Session Beans: In-Memory Replication





Inhalt

- Einführung BEA WebLogic
- J2EE 1.3
- WebLogic Java Application Server
- EJB 2.0: Home-Methods und Message Driven Beans
- EJB 2.0: Container Managed Persistence
- Mission critical Support für EJBs
- Zusammenfassung

Zusammenfassung

- Volles J2EE 1.3 heute verfügbar
 - Neu ist JCA, JMX
- EJB 2.0 ist mächtig
 - Mehr Struktur durch Home-Methods
 - EJB und JMS Hochzeit: Message Driven Beans
 - O/R Mapping durch Relationen
- Performance & Ausfallsicherheit durch BEAs Implementierung
 - Lazy Loading, verzögerte Updates
 - EJB In-Memory Replication im Cluster



How Business Becomes E-Business™