

Java Beans

Enterprise Java Beans

Eine kurze Einführung in die Welt
der Bohnen

Java Beans

Einführung

Stefan Sauer

Was ist ein Java Bean?

Beans sind Komponenten.

- Einmal schreiben
- Überall wiederverwerten
(und nicht jedesmal das Rad neuerfinden)
- Überall laufen (Plattformunabhängig)

Vier wichtige Teilgebiete:

- Introspektion (Selbstdarstellung, was hab ich, was kann ich)
- Persistenz (Speichern und wiederherstellen einer Bohne)
- Ereignisbehandlung (da war doch eine Mausbewegung)
- Eigenschaftenverwaltung (dazu jetzt mehr)

Was kann ich?

Beans bieten zu den benutzbaren Variablen
getter- und setter Methoden an.

```
public varType getVarName()  
public void setVarName(varType value)
```

Das geht auch mit Arrays. Aber anstatt ein ganzes
Array zu übergeben geht auch

```
public varType getVarName(int index)
```

Und warum nicht gleich public Variablen?
So kann beim setzen die Gültigkeit geprüft werden
und/oder eine Methode aufgerufen werden.

Bis jetzt war alles nur kalter Kaffee!

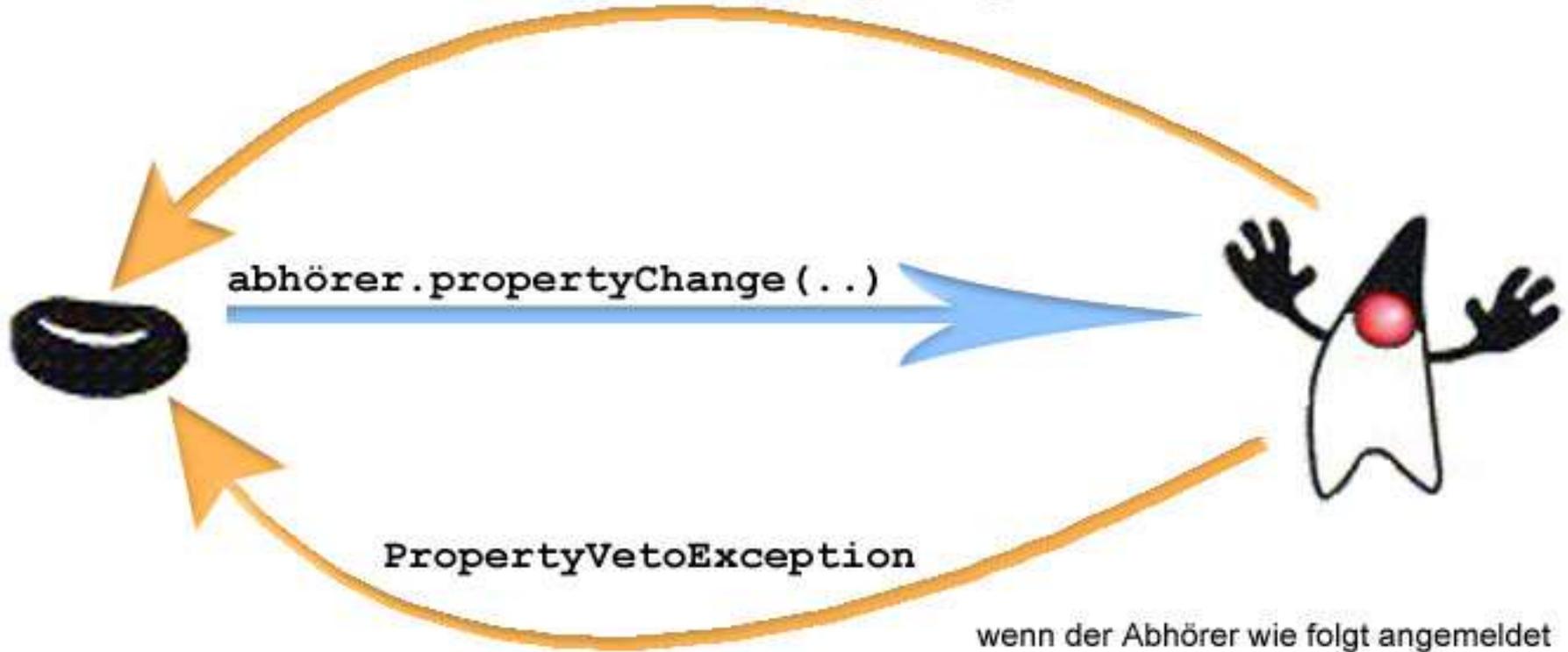
Um die Kommunikation zwischen Beans zu ermöglichen (und den Programmierer Arbeitslos zu machen) kann zu einer Variablen (oder auch Global) ein Abhörer (Listener) angemeldet werden.

Dieser wird sofort von der Bohne über den neuen Wert der überwachten Variable informiert.

Damit nicht genug, es gibt sogar Abhörer mit Veto-Recht. (Ne, der neue Wert gefällt mir nicht. Behalt mal den alten Wert)

So geht das Abhören

`bohne.addPropertyChangeListener(...)`



wenn der Abhörer wie folgt angemeldet
`bohne.addVetoableChangeListener(...)`

Enterprise Java Beans

Einleitung

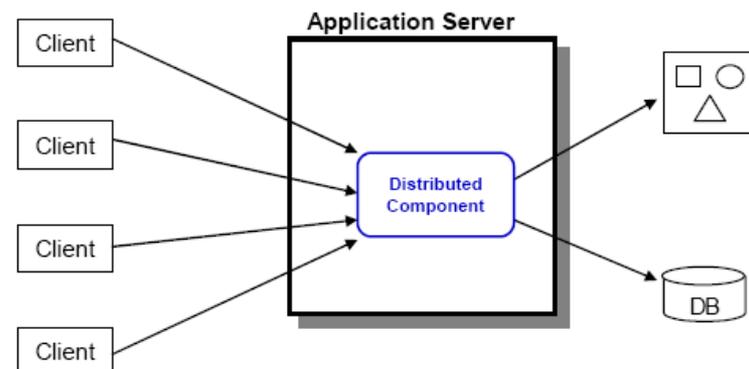
Sami Madani

Einleitung

- Wozu dienen Enterprise Java Beans?

Große Internetseiten erzeugen in der Regel einen enormen Rechenaufwand, bzw. müssen vielfältig verfügbar sein (z.B.: Internet-shop, Online-Banking, Google)

EJB dient zur Verwaltung und zur Kommunikation zwischen Clients – Datenbank (Distributed Framework)



Distributed Framework

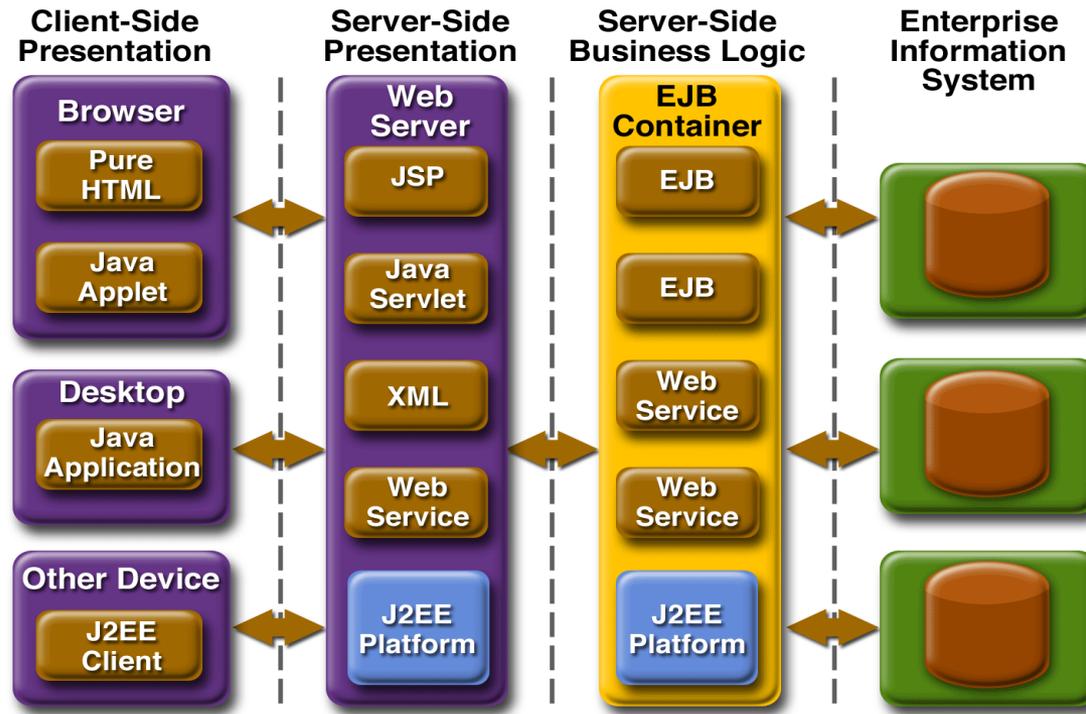
- Anforderungen an ein Distributed Framework:
 - Langlebigkeit / geringer Administrationsaufwand
 - niedrige Laufzeitkosten
 - Load-Balancing
 - geringer Entwicklungsaufwand
 - Plattformunabhängigkeit (ESB)

Realisierungsmöglichkeiten

- Lösungen existieren sowohl von Microsoft als auch von anderen Anbietern, die J2EE mit EJB und JSP sind der Lösungsvorschlag von Sun

Realisierung

- Realisierung mit EJB im Schichtmodell:



Das EJB-System

- EJB wird in einem sog. Modul-Container im EJBServer gestartet.

EJB-Server verwaltet:

- Kommunikation
- Load-Balancing
- u.a.

- Aufteilung im Container nochmals in Module, den Beans

Containereigenschaften:

- „Wohnort“ für die Beans
- sichere, erweiterbare Umgebung
- Handhabung des Lebenszyklus der Bean-Objekte

⇒ Geringer Entwicklungsaufwand, Bohnen sind vorgefertigt und müssen nur noch gefüllt werden

EJB Aufbau

- **EJB Home:**
 - Referenz zum EJB Container
 - Schnittstelle zum Client: Nur hier kann dieser einen neuen Bean erzeugen/löschen
- **EJB Object:**
 - nimmt Aufrufe des Clients entgegen
 - stellt den Beans die Umgebung bereit
 - ruft die eigentlichen Methoden auf
- **Enterprise Java Bean**
 - implementiert den eigentlichen Dienst, die Buiseness-Logic

Enterprise Java Beans

Bohnensorten

Jakub Lidke

Bohnensorten

- **Entity Beans sind Datenträger**
 - haben feste Attribute und können auf die DB zugreifen
- **Session Beans**
 - sind die Verbindung zum Client
- **Message Driven Beans**
 - sog. Message Listener, sie warten auf den Eingang von Nachrichten in Form von Objekten, Texten, usw.

Bohnsorten - Entity Beans

- Datenträger:
 - Kann mehrere Clients haben
 - Meist Repräsentant der Datenbank
 - Zugriff auf Attribute die in der DB gespeichert sind (get/set)
 - Modulation persistenter Geschäfts-Objekte
 - Hat ein Aktiven oder Passiven Zustand
 - Ueberlebt solange DB funktioniert (persistent)
 - Bean Managed – selbst Sicherung
 - Container Managed – Container sichert ab

Bohnensorten - Session Beans

- Buisness – Logic
 - 2 Typen Stateful (Applikation) und Stateless (Service)
 - nutzen ggf. andere Session/-Entity Beans
 - haben keine dauerhaften Attribute wie die Entity Beans
 - können Sessionsabhängig sein, z.B.: Einkaufswagen
 - bieten dem Client eine möglichst einfach Schnittstelle, indem sie vor anderen Beans gesetzt werden können um diese zu Kapseln
 - kann aus DB lesen und diese updaten
 - Hat ein Aktiven oder Passiven Zustand
 - Der Diener des Clients auf dem Server

Bohnsorten - Message Driven Beans

- **Message Listener:**
 - Werden asynchron vom Container aufgerufen
 - Warten auf den Eingang von Nachrichten an einem bestimmten Zielort
 - anonym fuer den Client – keine ID nur ein Zielort bekannt
 - Alle MDB-Klassen muessen das MessageDrivenBean und ein passendes MessageListener interface implementieren.
 - Session- und EntryBeans duerfen keine Listener sein.

EJB-Client - Was kann er?

- Objektreferenzen des Home-Beans ueber JNDI ermitteln
- SessionBean-Instanz erzeugen
- Methoden aufrufen
- Bean-Instanzen loeschen
- Die Transaktion starten und beenden
- Methoden dynamisch aufrufen
- Handle serialisieren (Speichern einer Session Referenz)

Code Beispiele – Client Zugriff

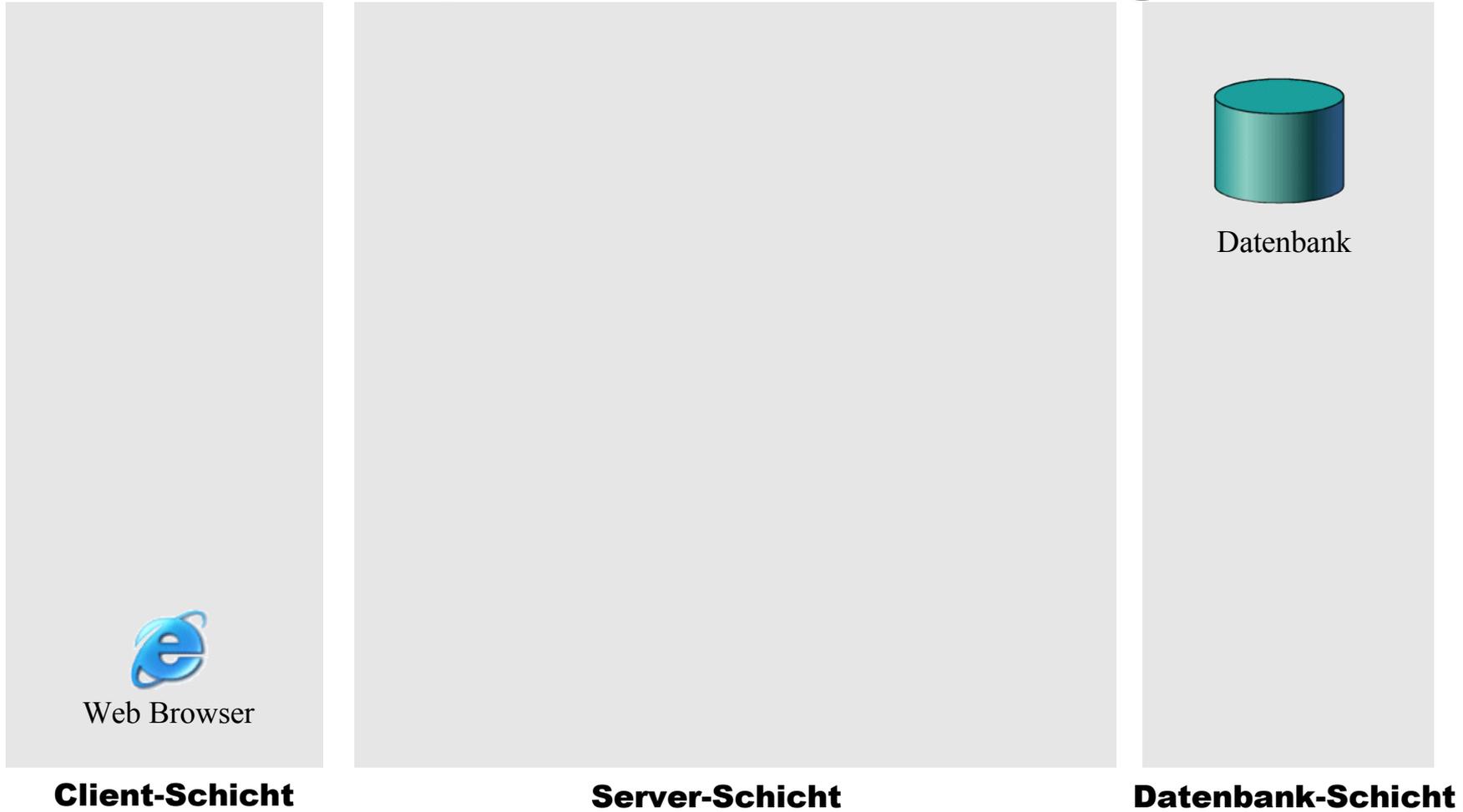
```
// Objektreferenz des EJBHome-Objektes über JNDI ermitteln
Context ctxt = new InitialContext();
AccountHome home =(AccountHome)ctxt.lookup("Sparkasse");
// Eine neue EJB-Instanz erzeugen
Account bean = home.create("Hugo");
// Methoden aufrufen
bean.deposit(100);
...
// Bean löschen
bean.remove();
```

Anwendungsbeispiel

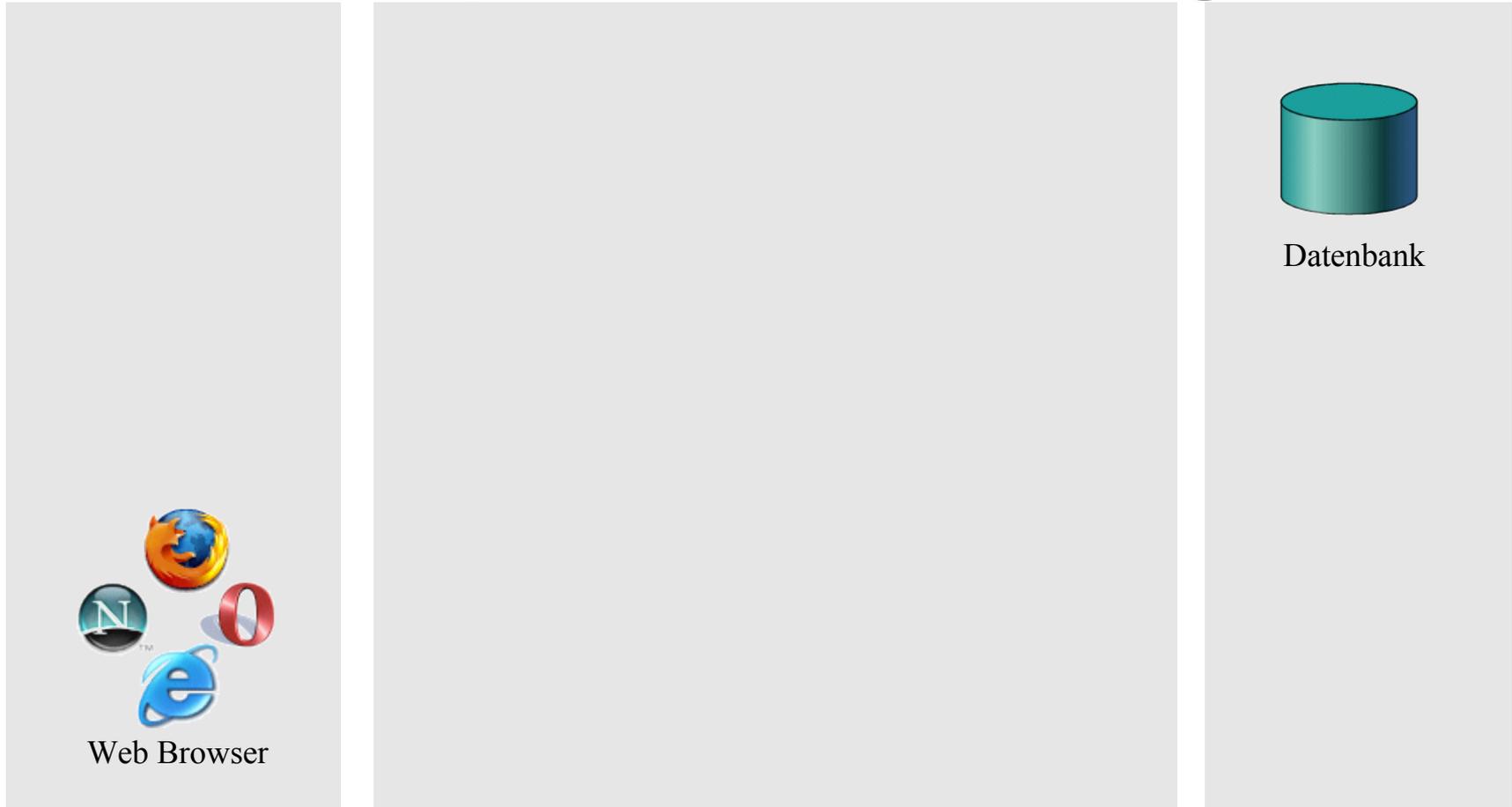
Was bringen uns
Enterprise Java Beans?

Manuel Haim

Beispiel: Homebanking



Beispiel: Homebanking



Client-Schicht

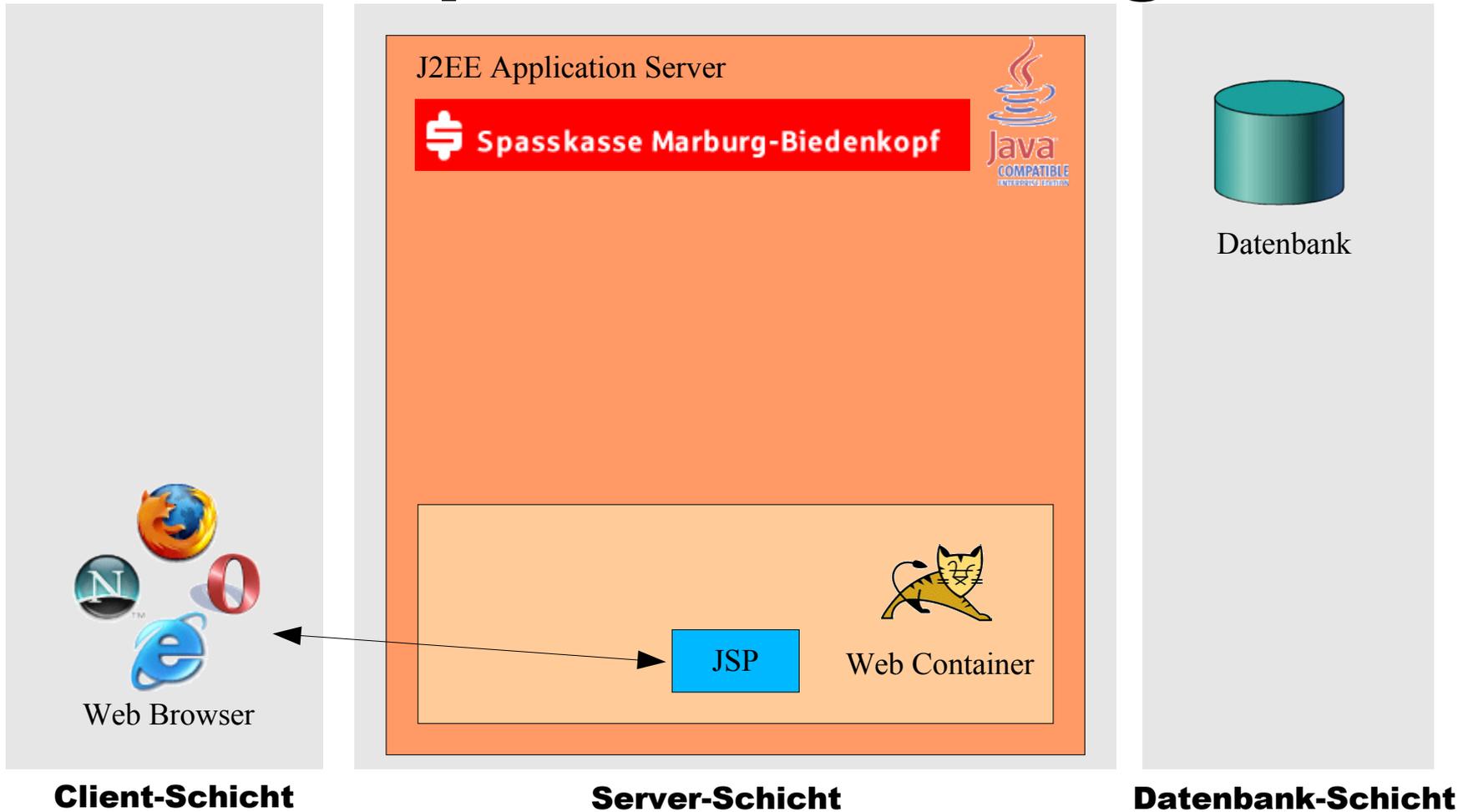
Server-Schicht

Datenbank-Schicht

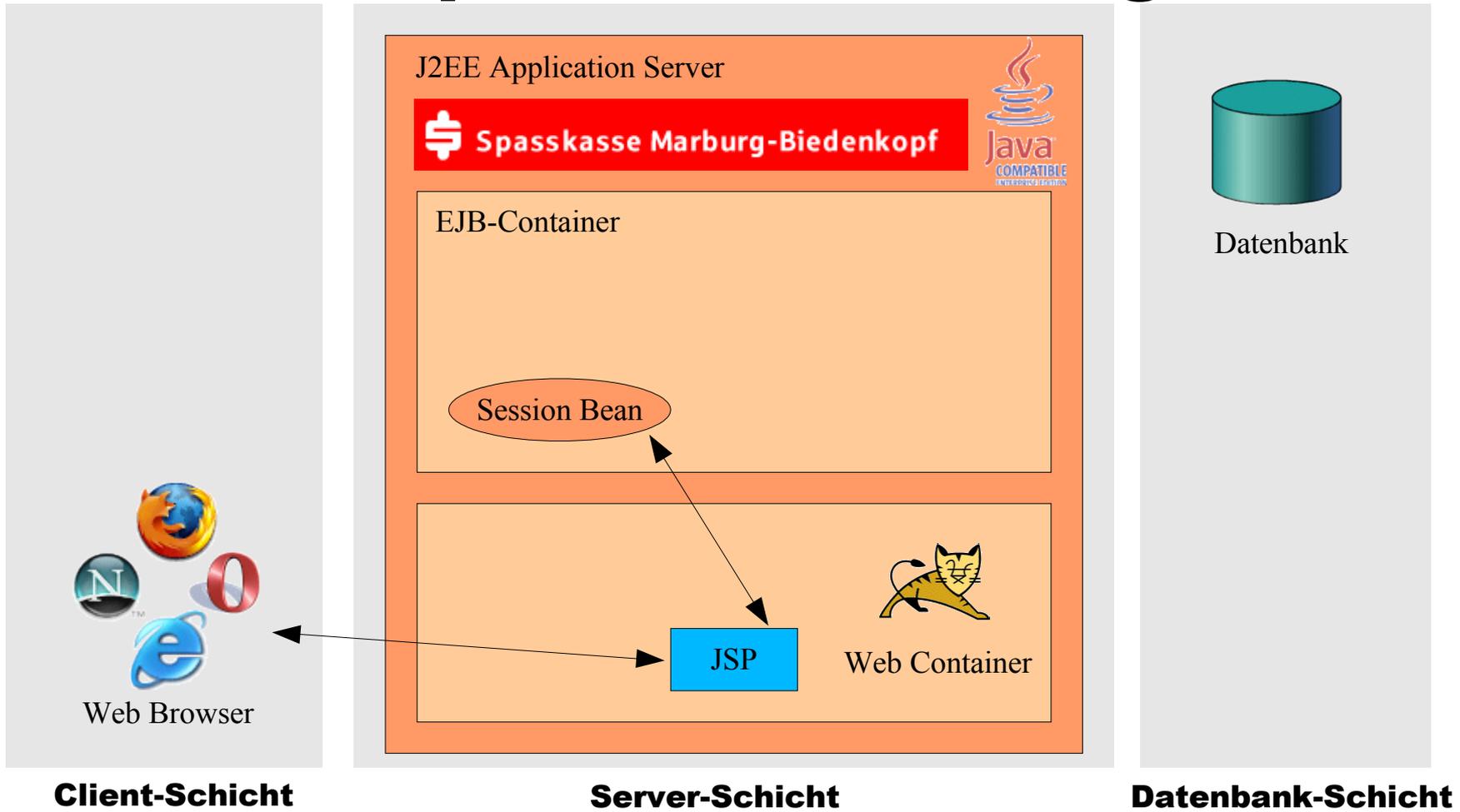
Beispiel: Homebanking



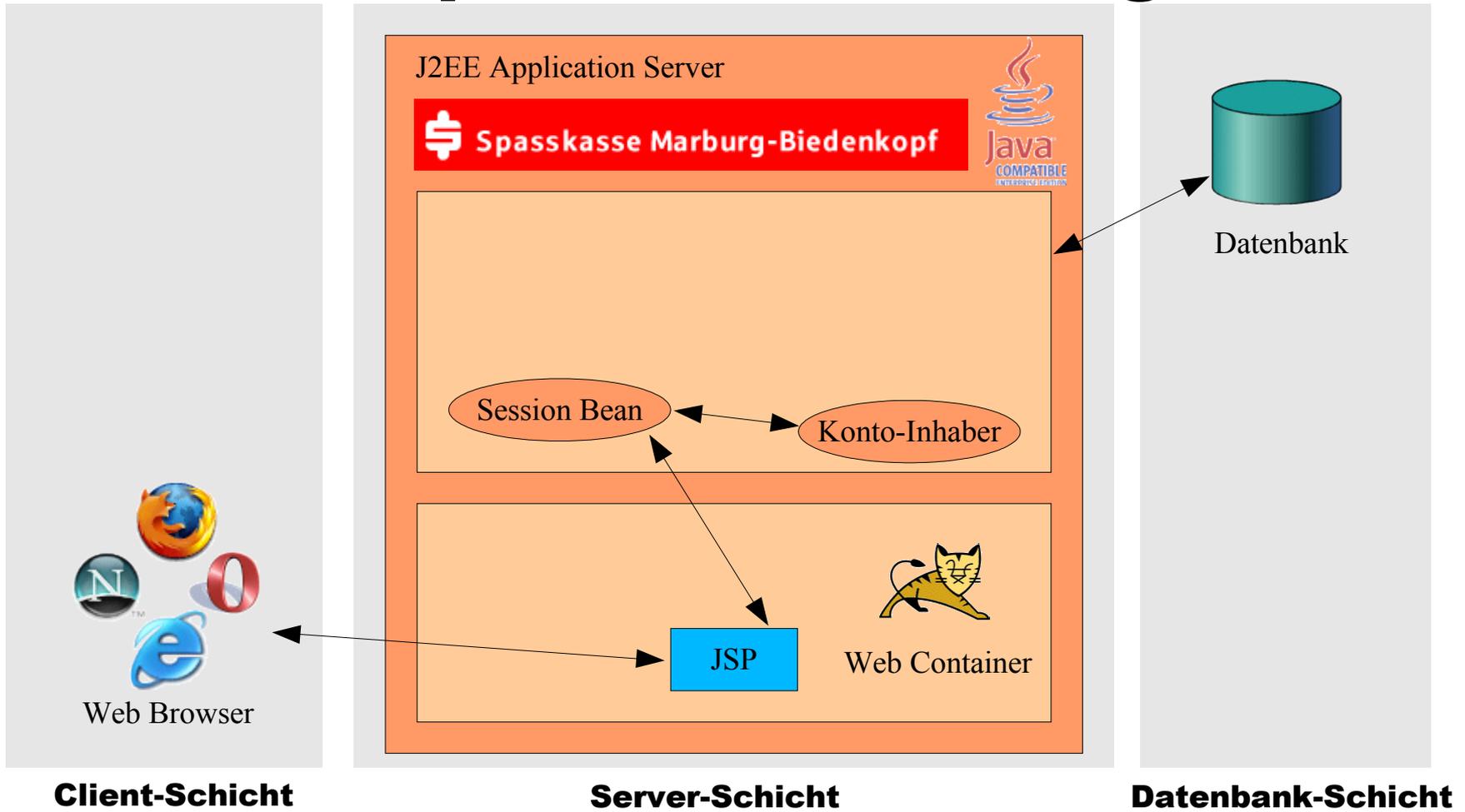
Beispiel: Homebanking



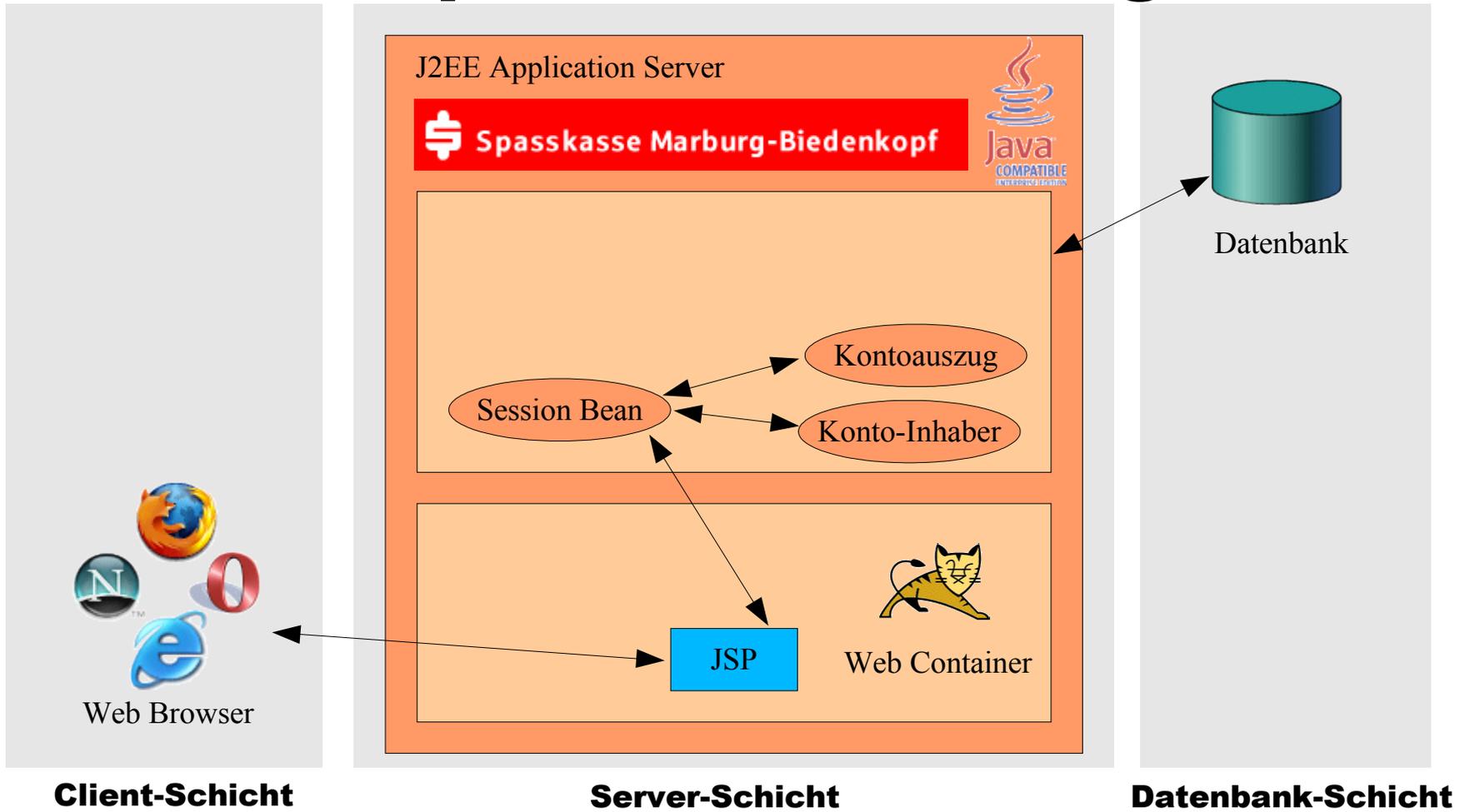
Beispiel: Homebanking



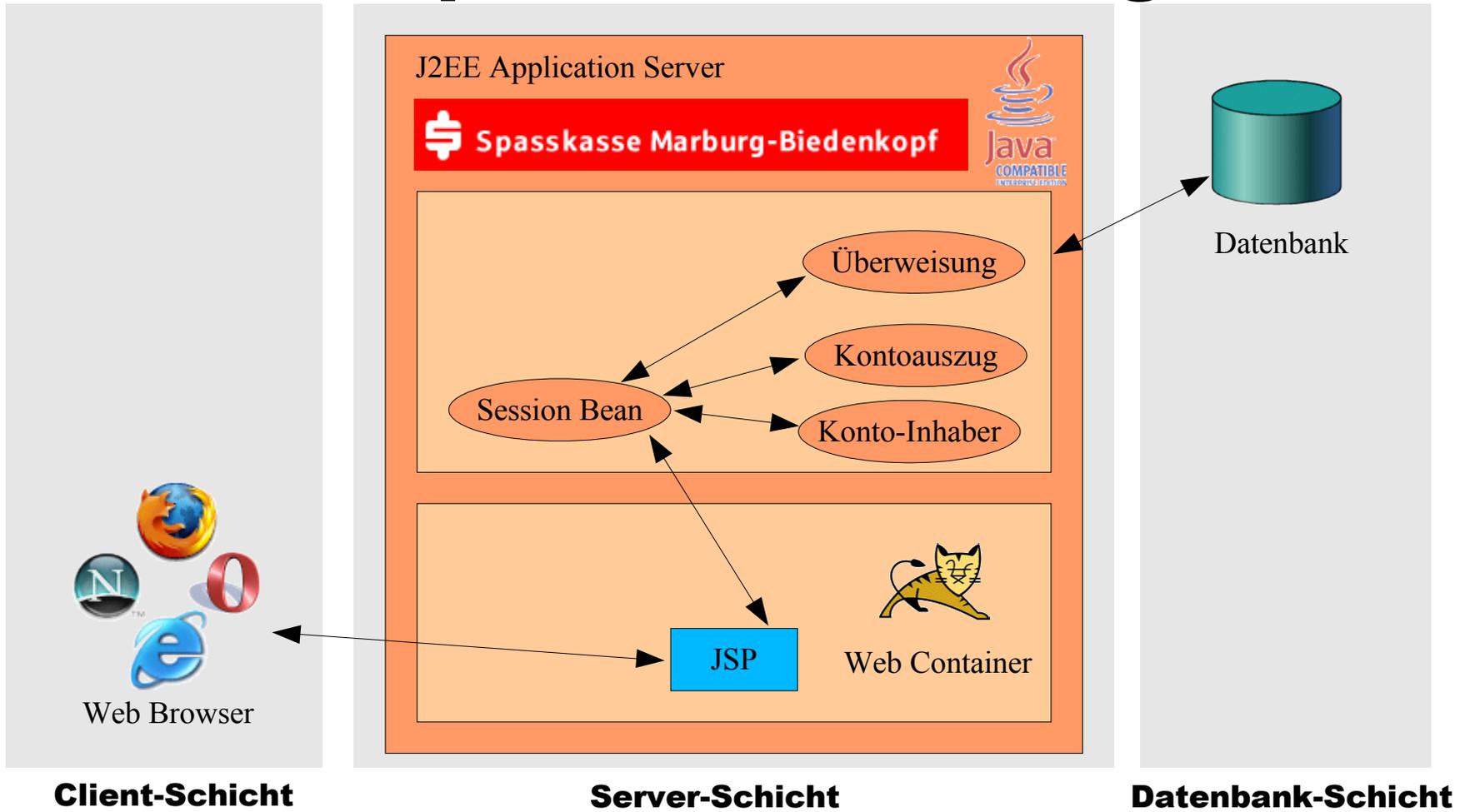
Beispiel: Homebanking



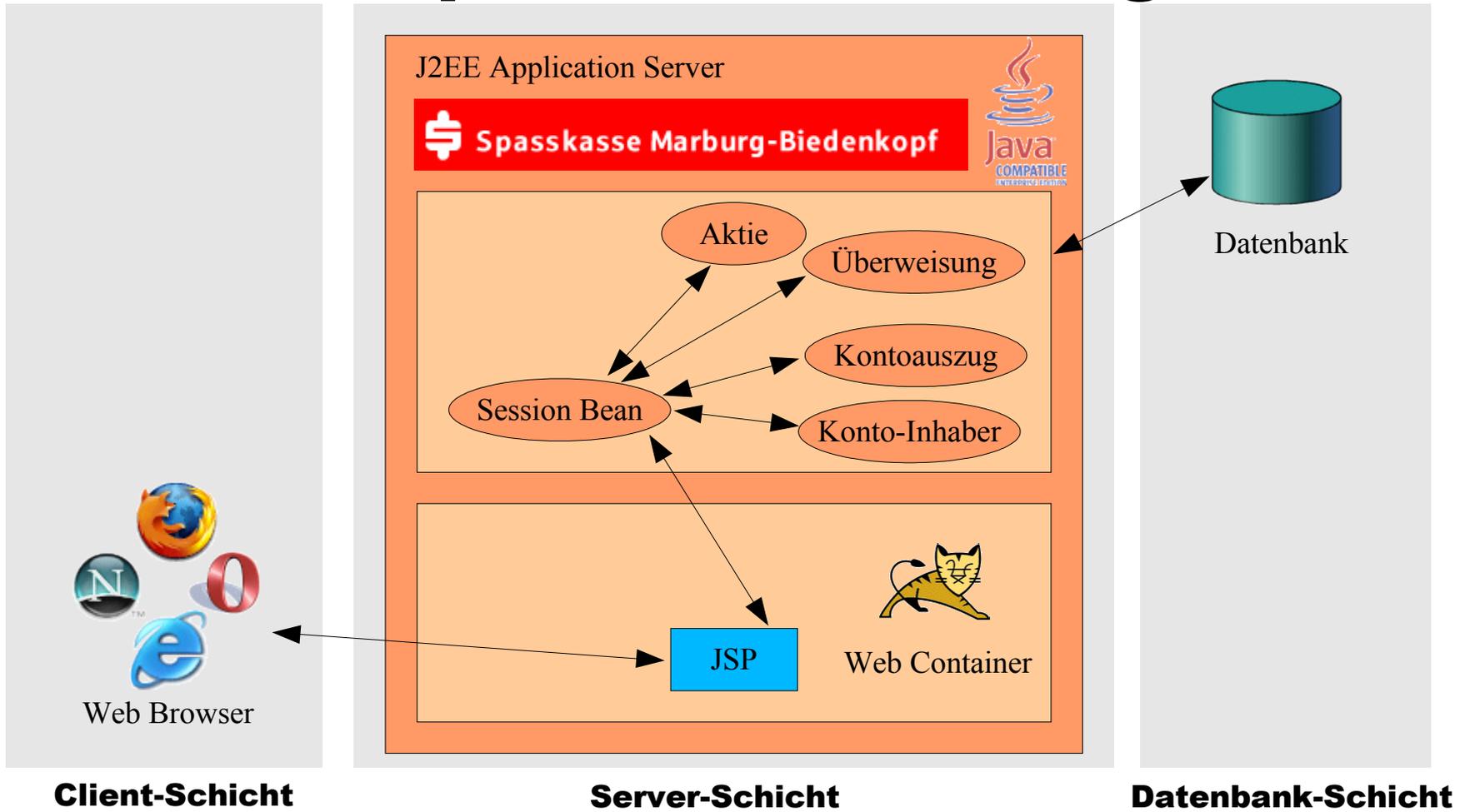
Beispiel: Homebanking



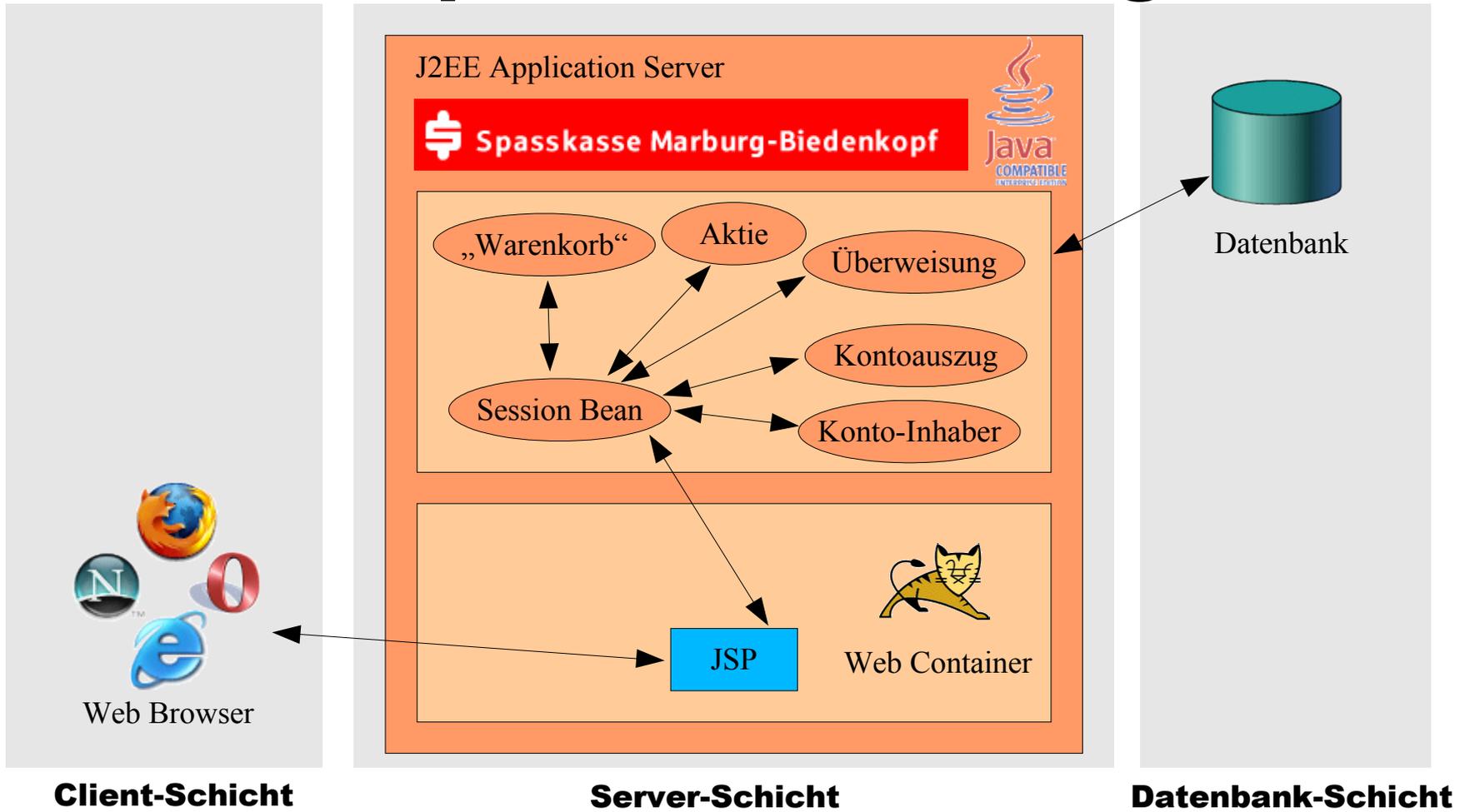
Beispiel: Homebanking



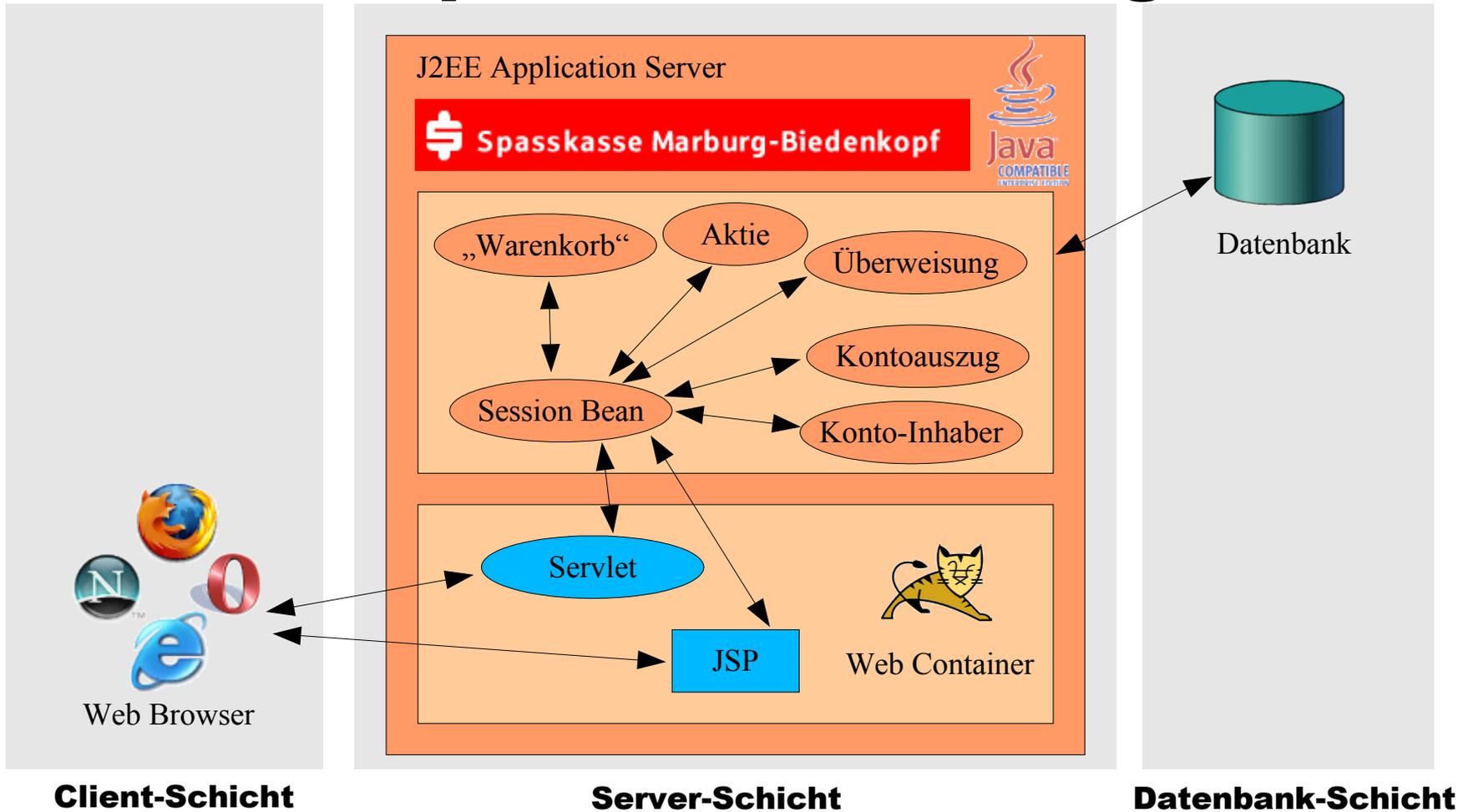
Beispiel: Homebanking



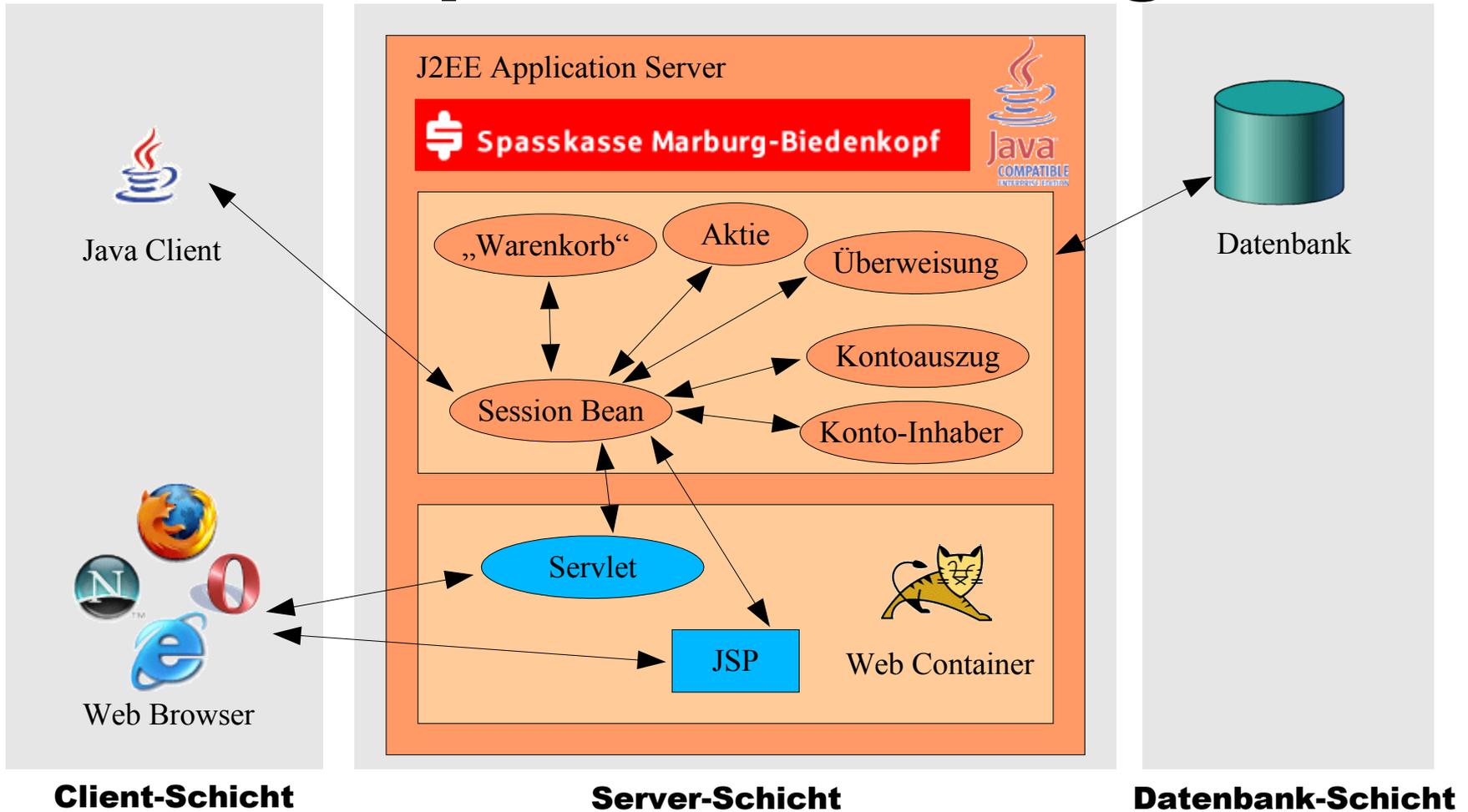
Beispiel: Homebanking



Beispiel: Homebanking



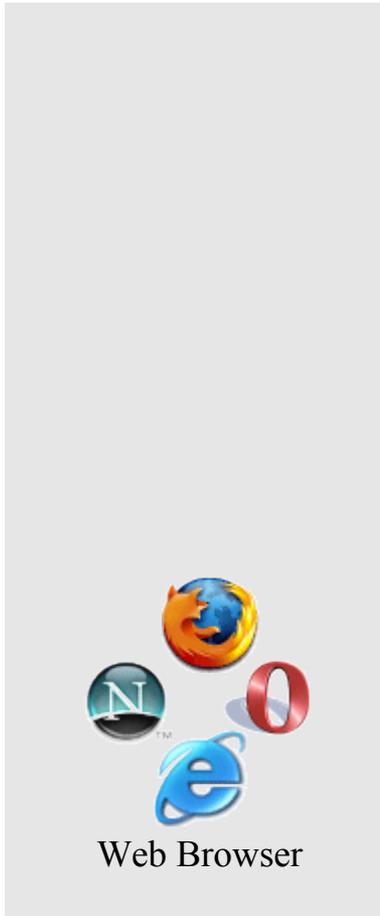
Beispiel: Homebanking



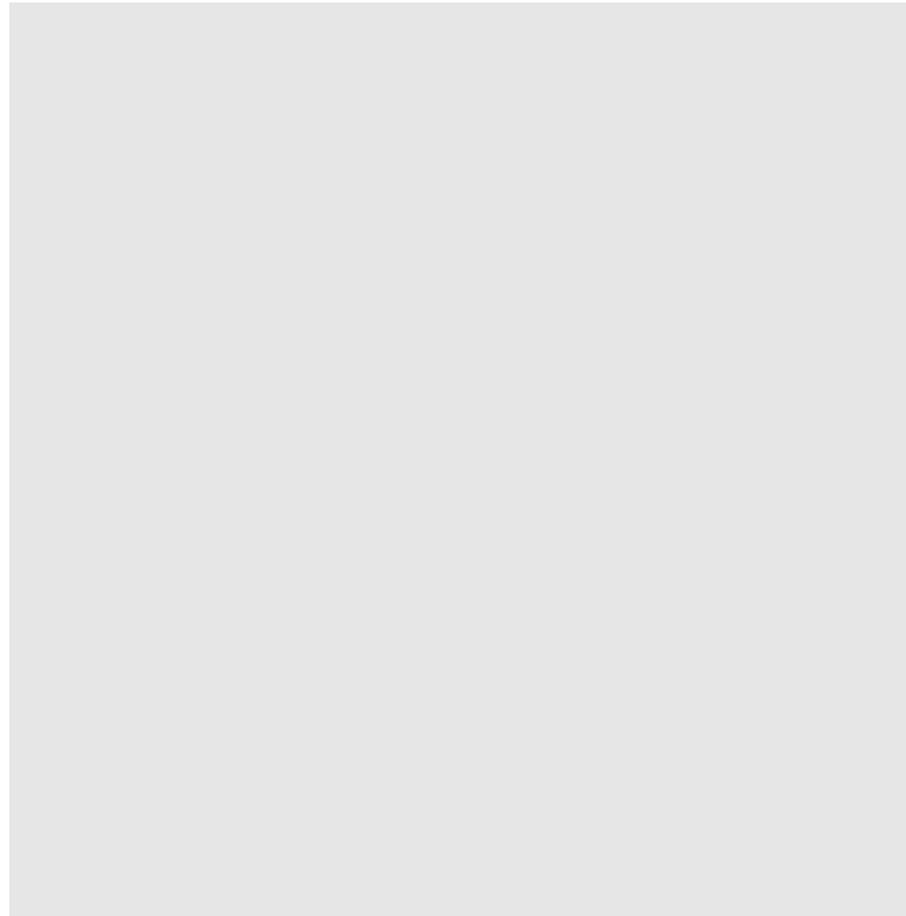
Frage:

Was bringen
Enterprise Java Beans
für uns?

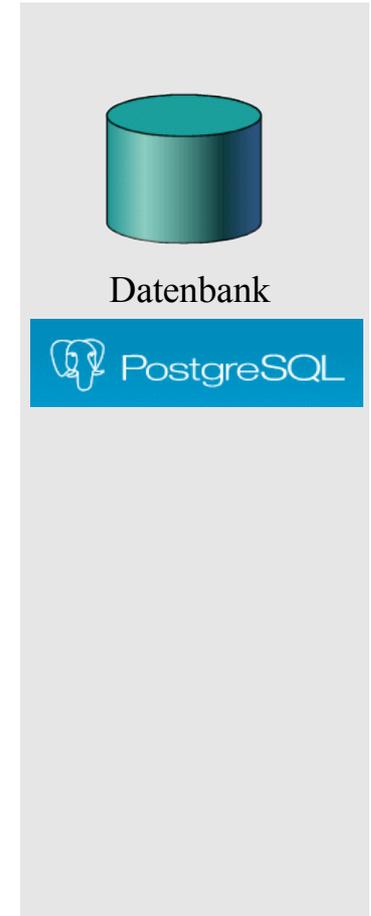
Literaturdatenbank



Client-Schicht

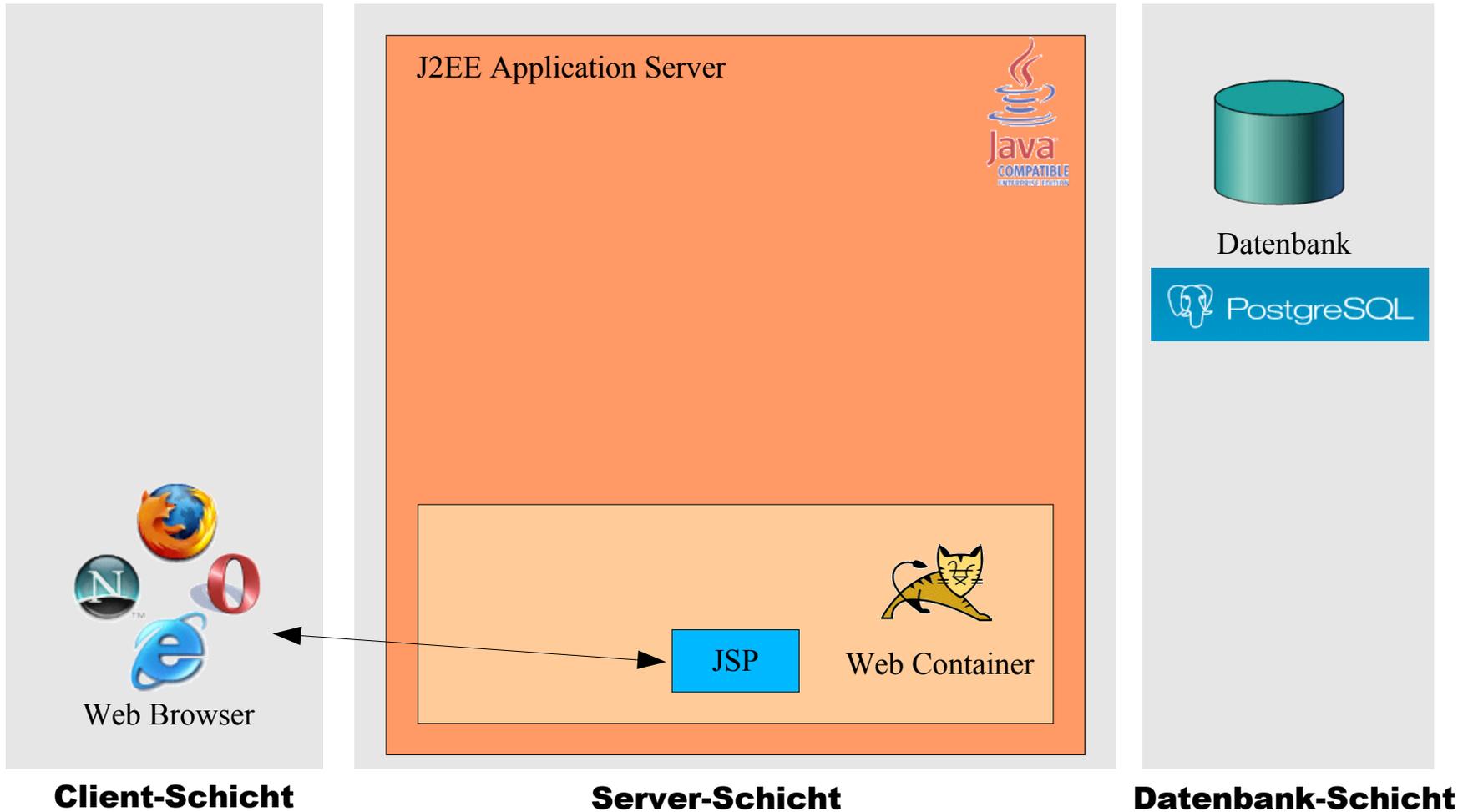


Server-Schicht

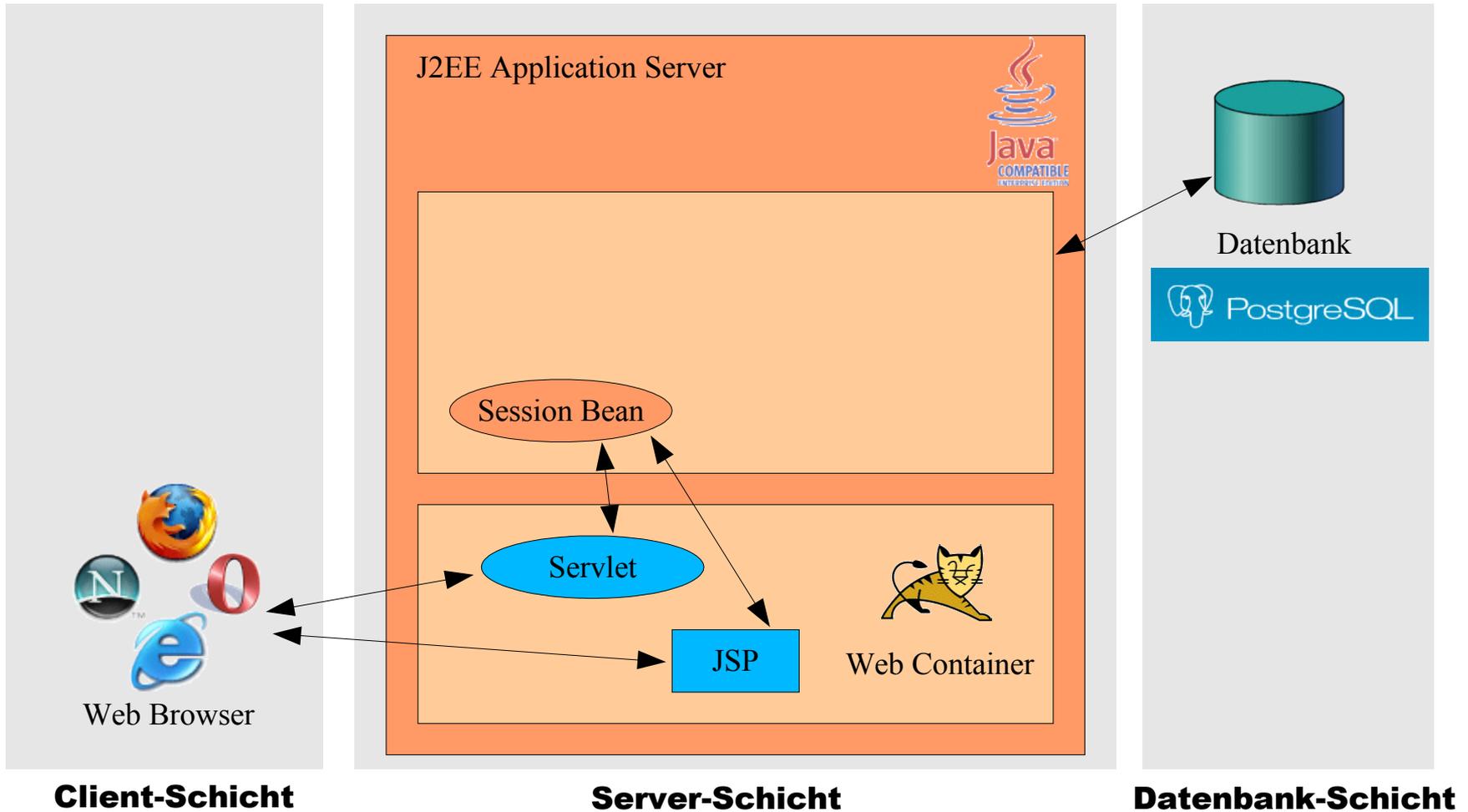


Datenbank-Schicht

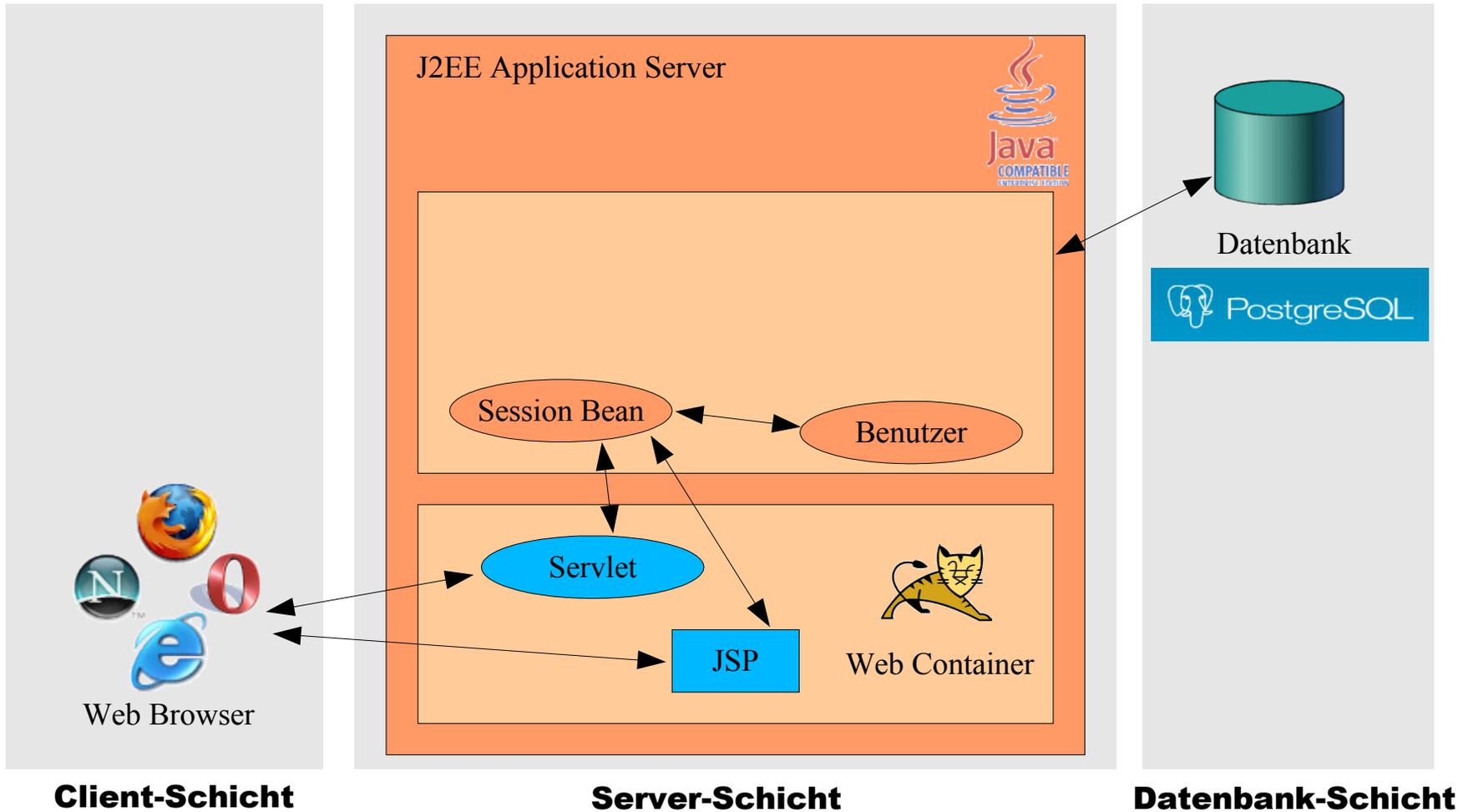
Literaturdatenbank



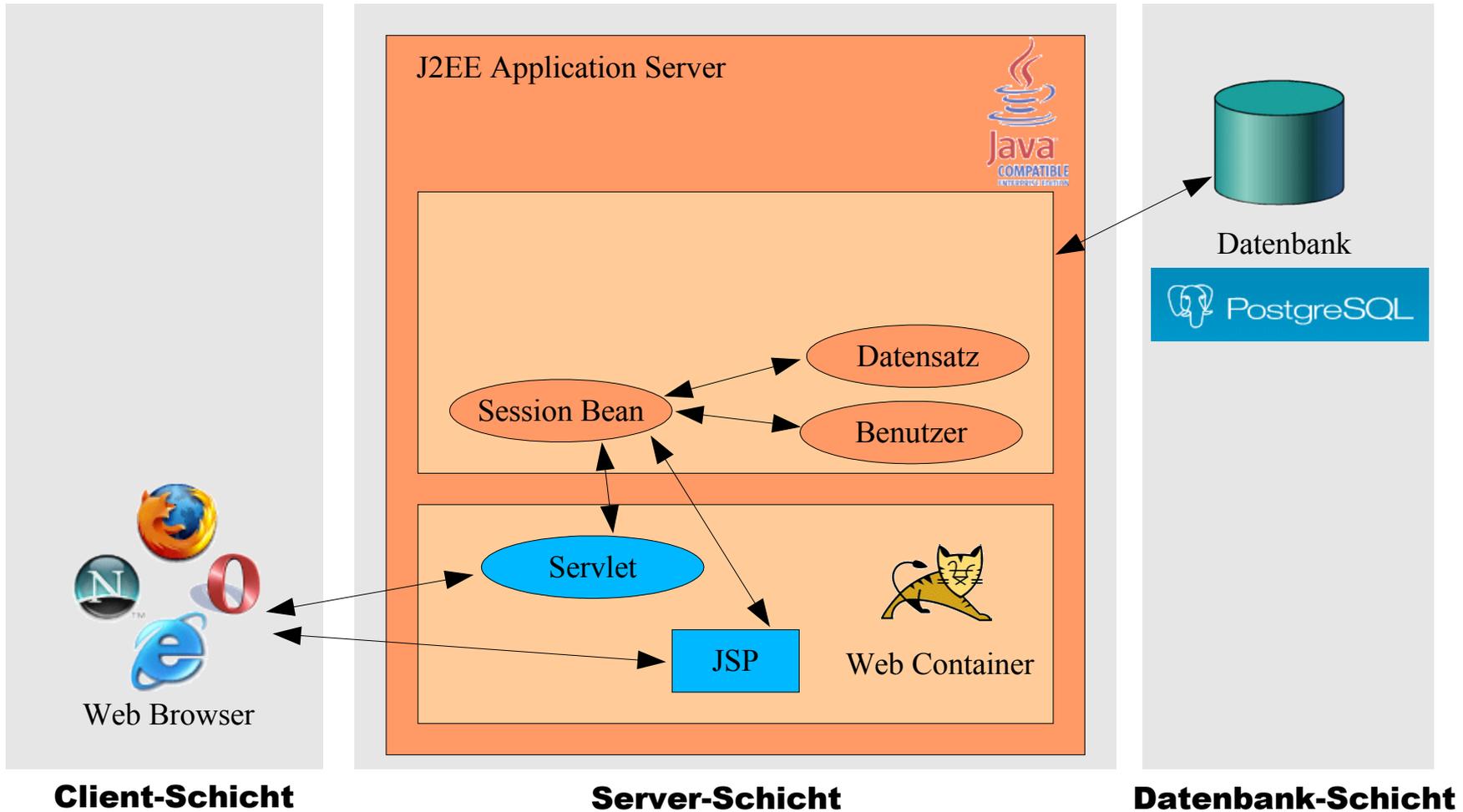
Literaturdatenbank



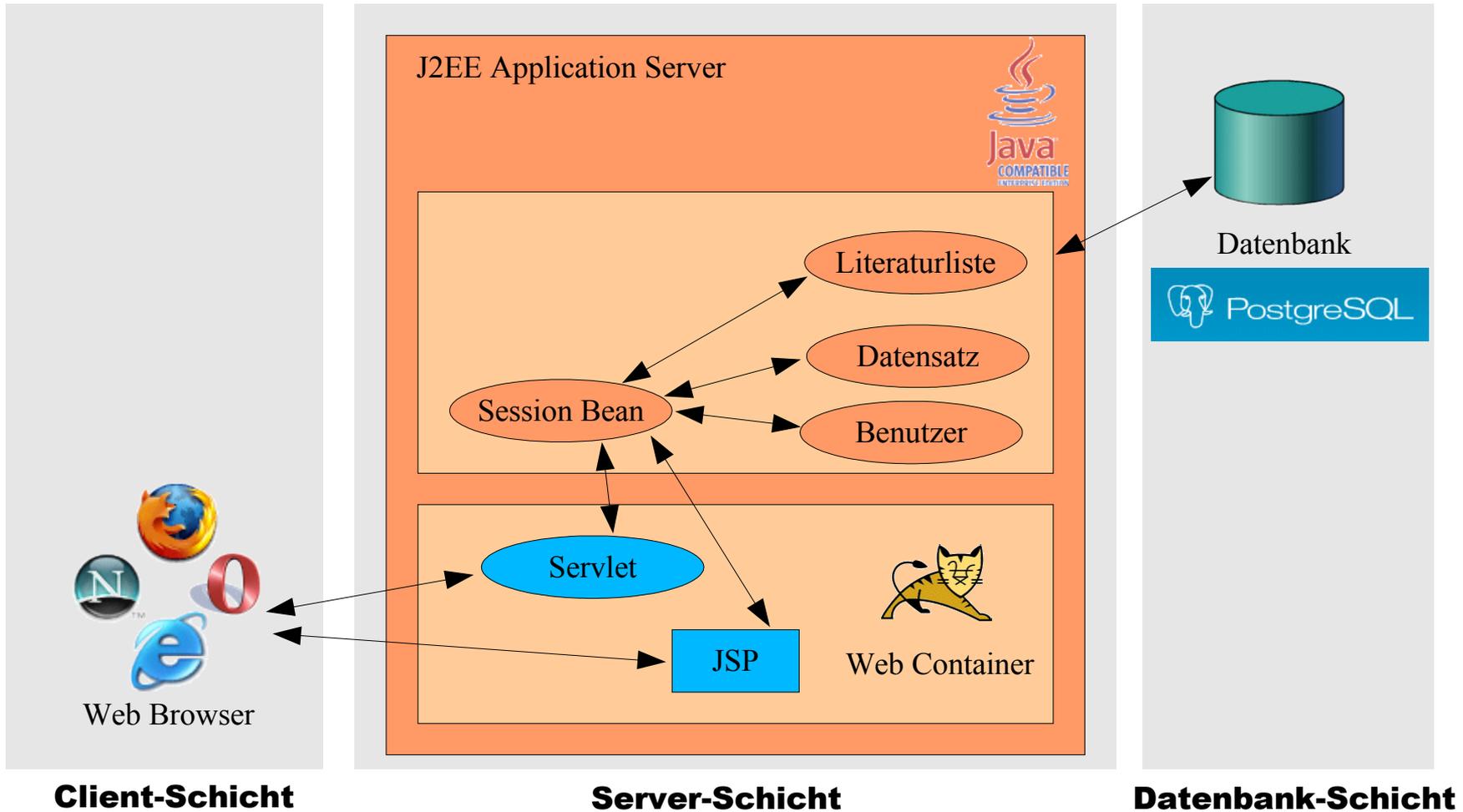
Literaturdatenbank



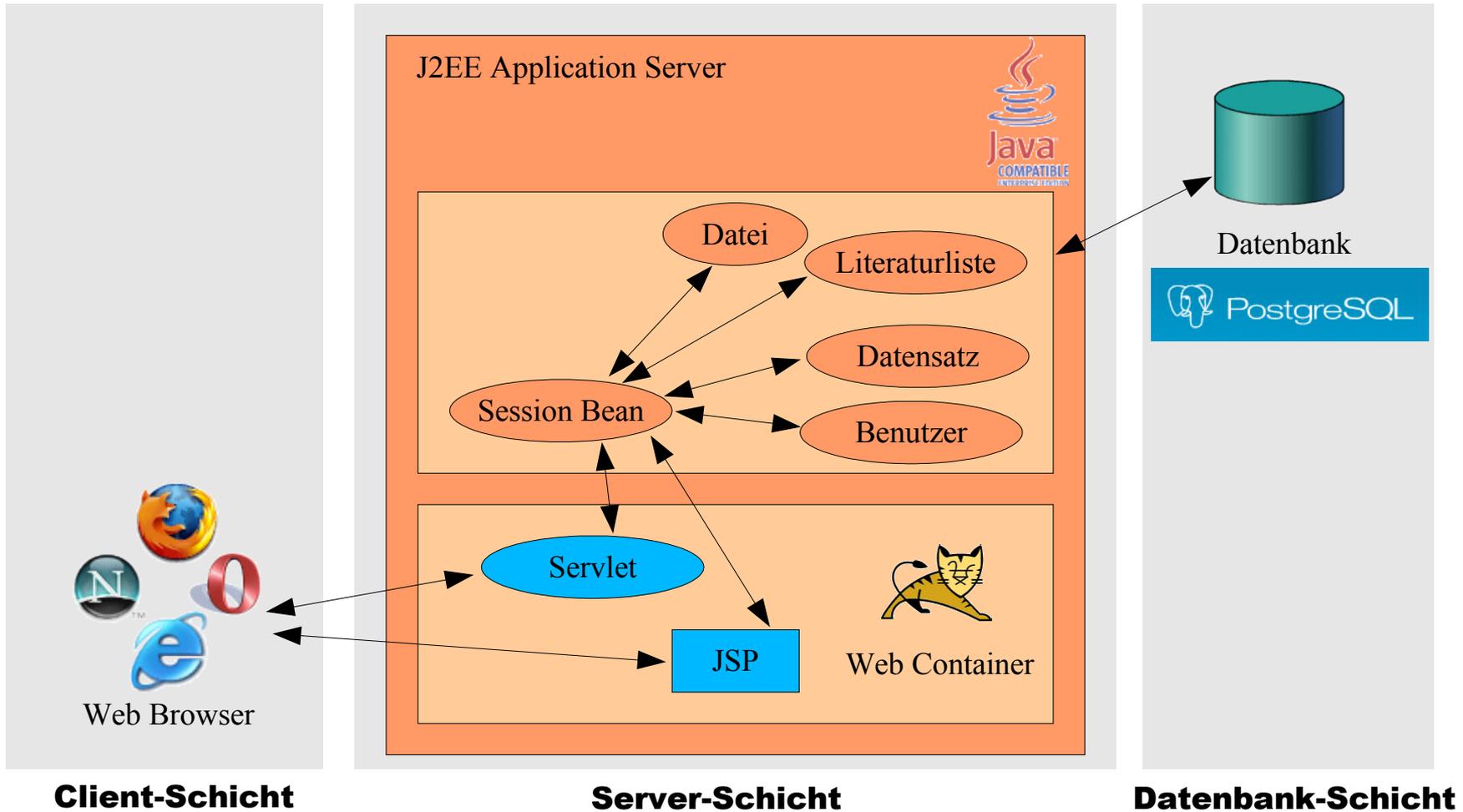
Literaturdatenbank



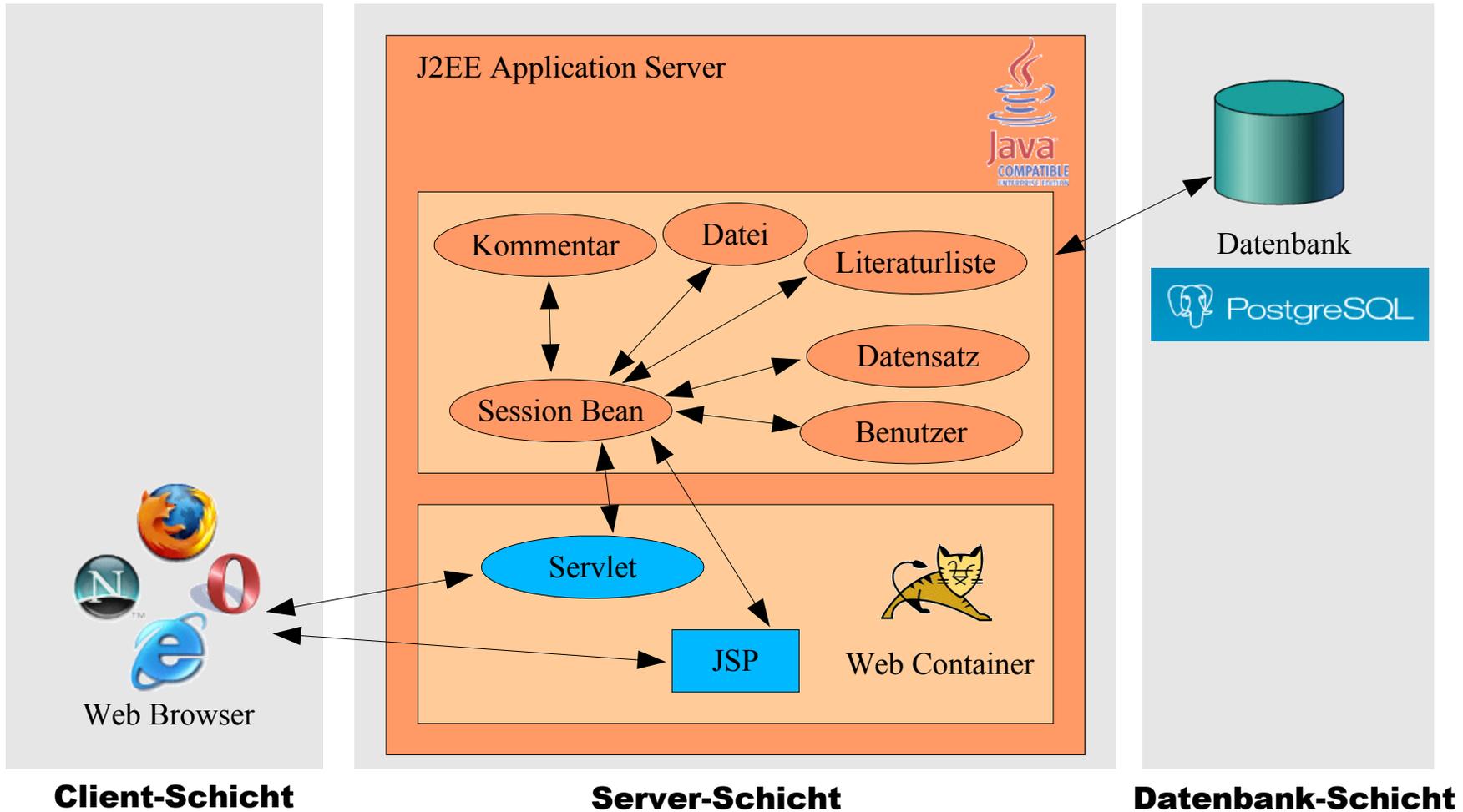
Literaturdatenbank



Literaturdatenbank



Literaturdatenbank



Und jetzt? Wochenende!

Quellen:

<http://java.sun.com>

http://www.inform.fh-hannover.de/file_doku/ejbForGiBraunschweig.pdf

<http://www.theserverside.com/books/wiley/masteringEJB/>