



OPITZ CONSULTING



# Oracle Business Components for Java

Sven Willmann / OPITZ CONSULTING  
[www.opitz-consulting.de](http://www.opitz-consulting.de)



- ▲ **J2EE Applikationen**
- ▲ **Oracle Business Components for Java**
  - **Architektur und grundlegende Konzepte**
  - **Implementierung von Geschäftslogik**
  - **Verteilungsmöglichkeiten**
  - **Demo**
- ▲ **Diskussion**



## ▲ J2EE

- Technologien und API's stellen die Zutaten zur Verfügung um Enterprise Applikationen zu entwickeln.
- Diese Applikationen können dann zu einem J2EE konformen Application Server deployed werden.

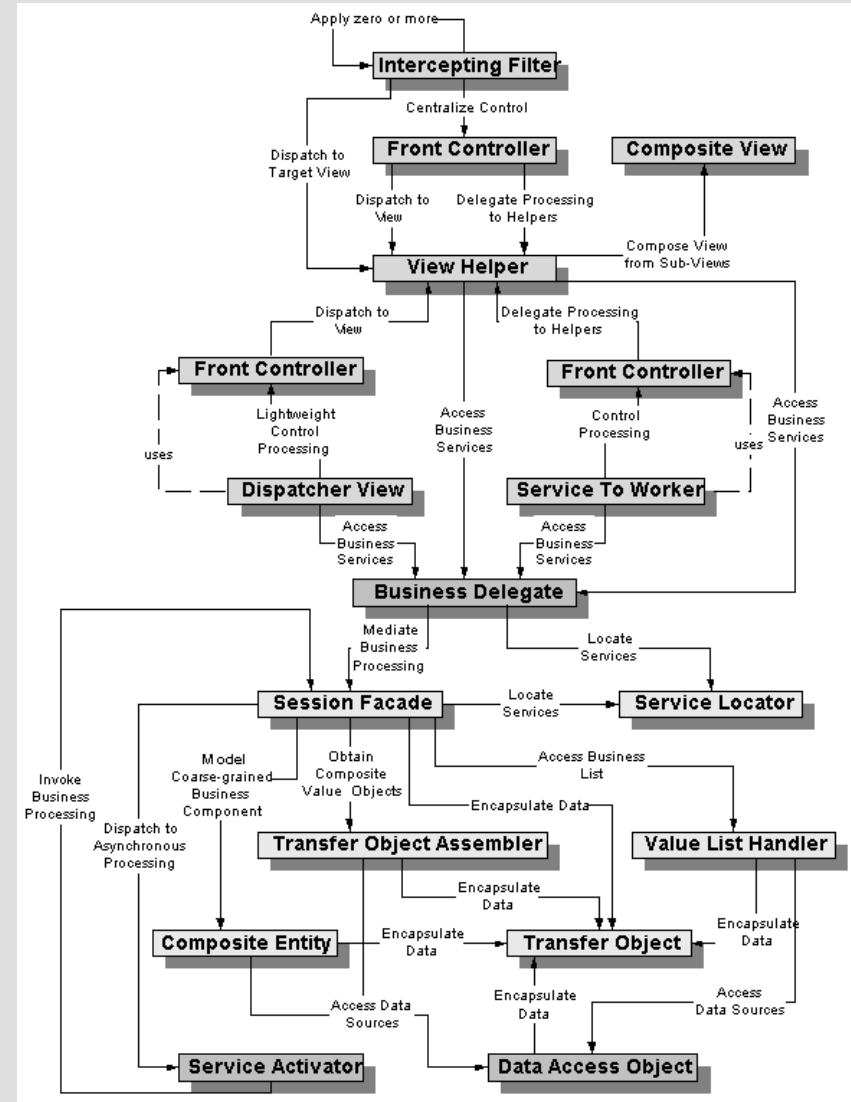
## ▲ Ein großer Unterschied im Aufwand besteht zwischen

- erstellen einer J2EE konformen Applikation und
- einer, welche modular, performant und skalierbar ist.



▲ Wie kann man modulare, performante und skalierbare J2EE Applikationen entwickeln

- Design Patterns verstehen und immer wieder auf's neue anwenden.
- Basisfunktionalitäten übergreifend durch wiederverwendbare Komponenten bereitstellen.
- **Fertige Komponenten nutzen!**





**Oracle Business Components for Java (BC4J) ist ein technisches Framework basierend auf Java und XML.**

**Es bietet:**

- ▲ **produktive Entwicklung der mittleren Schicht von mehrschichtigen, datenbank-getriebenen Java Applikationen,**
- ▲ **flexible Verteilungsmöglichkeiten für diese Komponenten der mittleren Schicht,**
- ▲ **leichte Anbindung von schmalen Java Swing-, Servlet- und JavaServer Page Clients etc.**



- ▲ **BC4J implementiert J2EE design patterns**
- ▲ **Damit wird die Verantwortung des "richtigen" Implementierens vom Entwickler genommen.**
- ▲ **Folglich:**
  - Einfache Entwicklung von Applikationen
  - Verkürzte Entwicklungszeit
  - Hohe Funktionalität
  - Erweiterbarkeit
  - Einheitlichkeit
  - Hohe Qualität



**Technische Frameworks unterstützen die Anwendungsentwicklung.**

**Sie umfassen:**

- ▲ **Softwarebausteine mit Basisfunktionalitäten wie Connection Pooling, Caching, Locking, Master-Detail-Synchronisation**

***BC4J: Java Klassenbibliotheken***

- ▲ **Tool-Unterstützung bei der Softwareentwicklung**

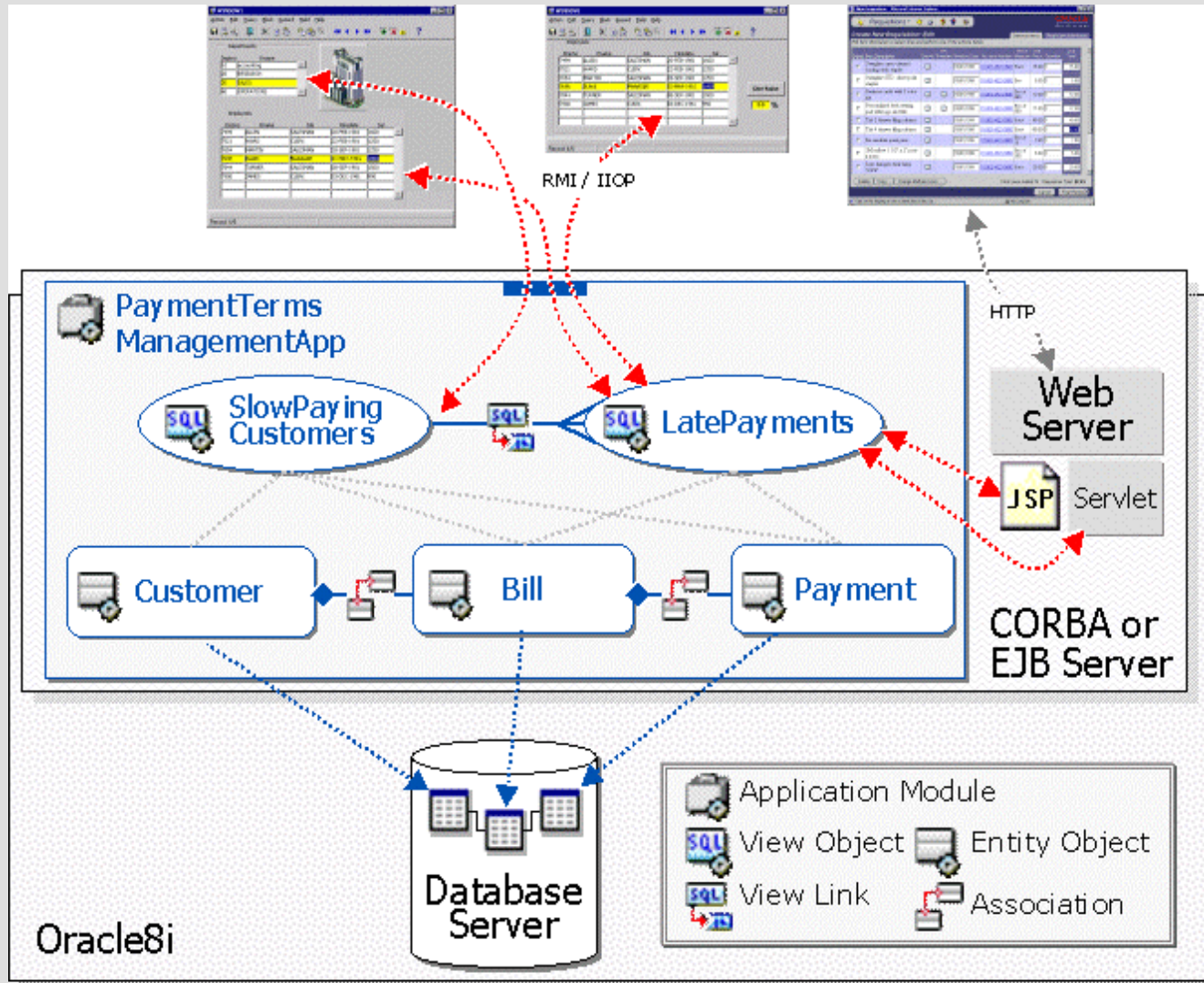
***BC4J: Oracle9i JDeveloper***

**Im Gegensatz dazu *fachliche Frameworks*:**

- ▲ **Vordefinierte Geschäftskomponenten für bestimmte Einsatzbereiche**

- ▲ **Zum Beispiel: Adressverwaltung, Lagerverwaltung, Buchhaltung...**

**BC4J ist kein fachliches Framework, entsprechende Komponenten können aber mit BC4J entwickelt werden.**



Quelle: Oracle



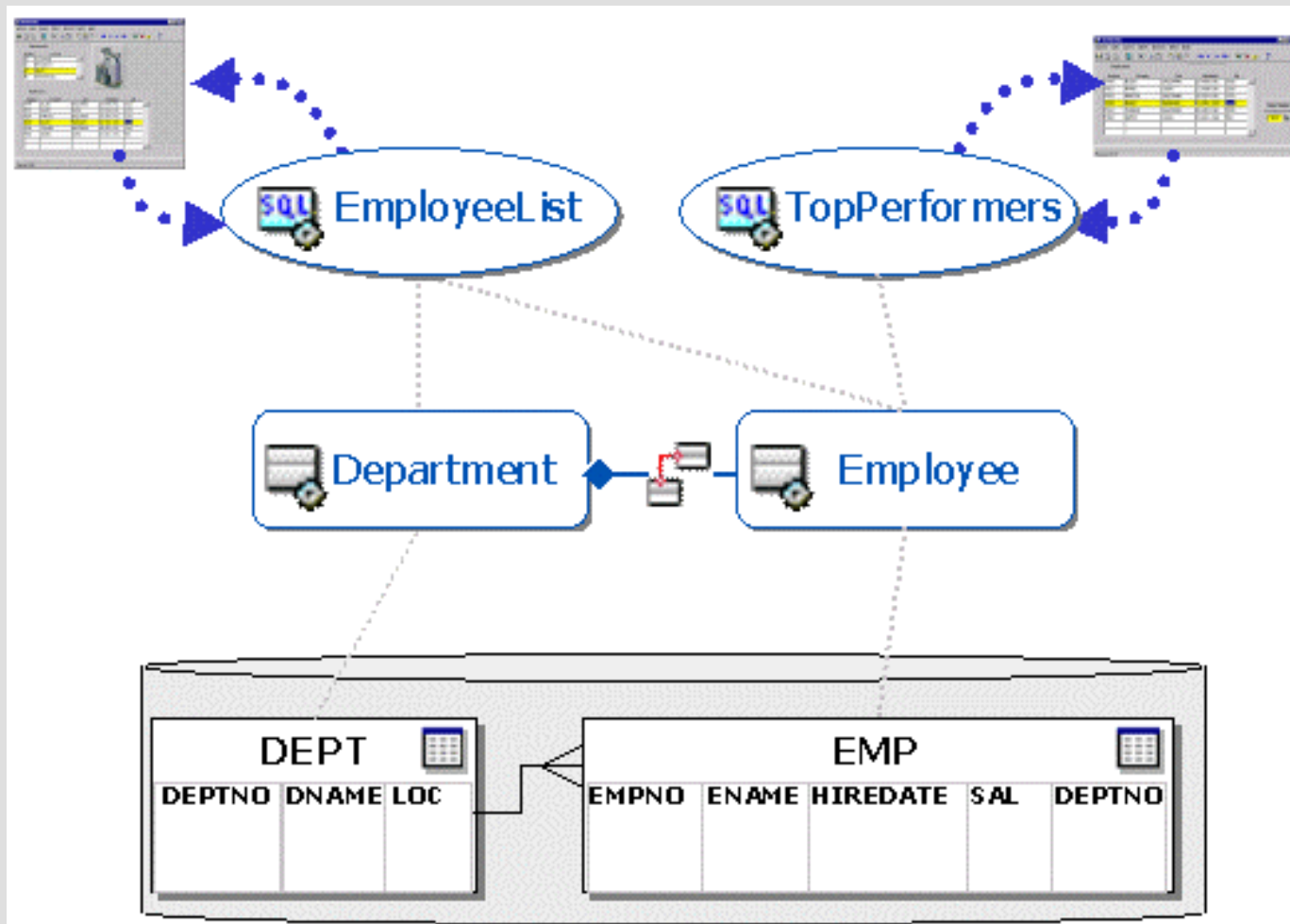
## Entitiy Objects (EOs):

- ▲ EOs entsprechen in der Regel den Tabellen der Datenbank. Spalten der Tabellen werden dabei zu Attributen der EOs.
- ▲ Über EOs wird u.a. Persistenz und Caching realisiert. Ferner kann hier zentral Geschäftslogik implementiert werden.
- ▲ Ein EO besteht aus einer XML-Datei (Metadaten, Deklaration von Geschäftslogik) und einer Java Datei (Implementierung von Geschäftslogik).
- ▲ Beziehungen zwischen EOs werden durch Assoziationen abgebildet. Assoziationen beruhen z.B. auf Foreign Keys der zugehörigen Tabellen.



### **View Objects (VOs):**

- ▲ **VOs steuern den Zugriff auf die Daten der Applikation. Dazu kapseln sie SQL SELECT Anweisungen.**
- ▲ **VOs können auf EOs beruhen, beim Ausführen der Abfrage eines VO werden über den Cache die zugehörigen Entitys erzeugt. (Ausnahme: Read Only VOs)**
- ▲ **Clients greifen nur über VOs auf EOs zu. VOs bieten dazu alle benötigten Funktionen zum Navigieren in Ergebnismengen.**
- ▲ **Auch Änderungen geschehen über die Attribute von VOs. Dies wird dann an die zugrundeliegende EOs mit ihrer Geschäftslogik delegiert.**
- ▲ **Master-Detail-Beziehungen werden über View Links koordiniert.**

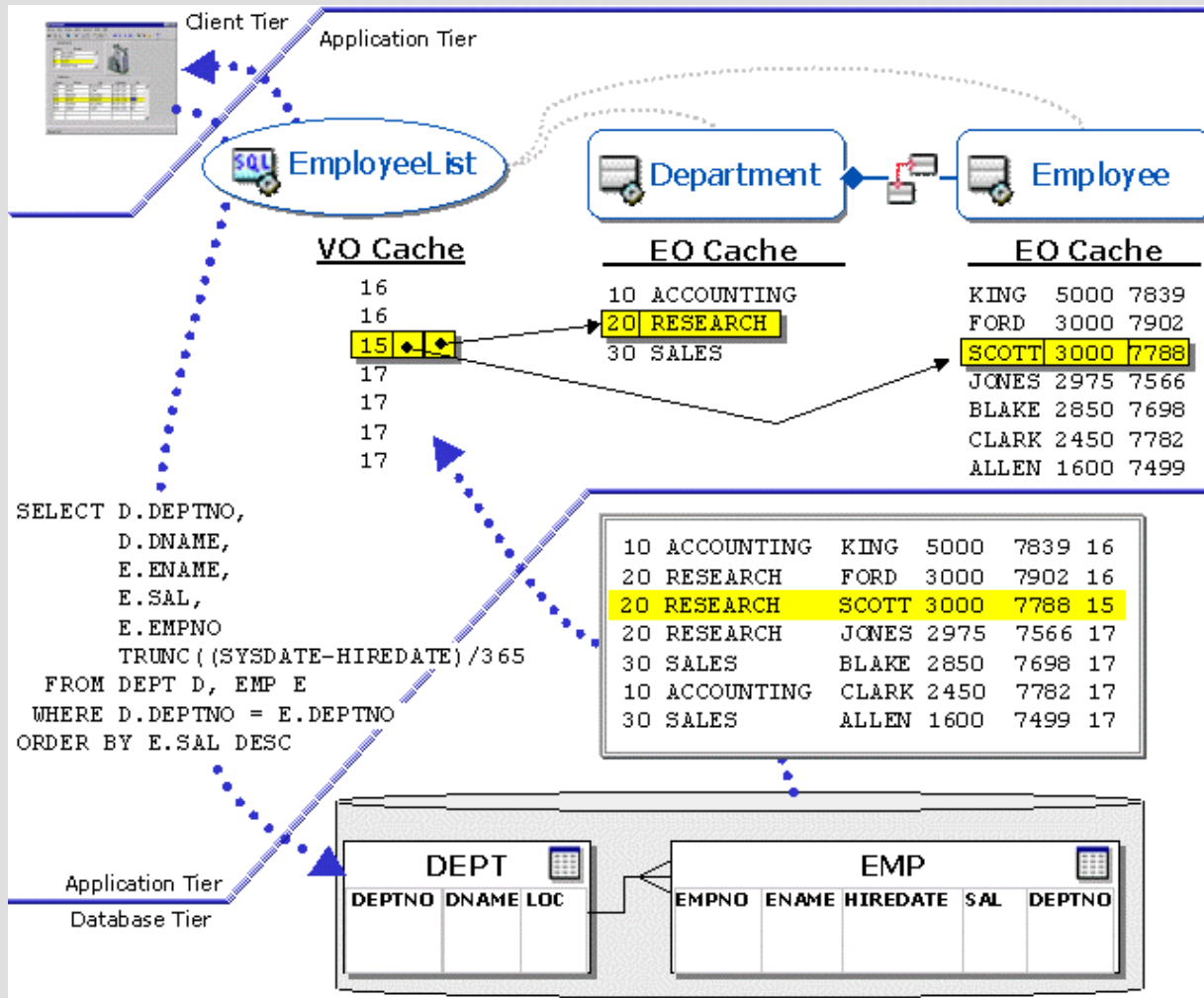


Quelle: Oracle

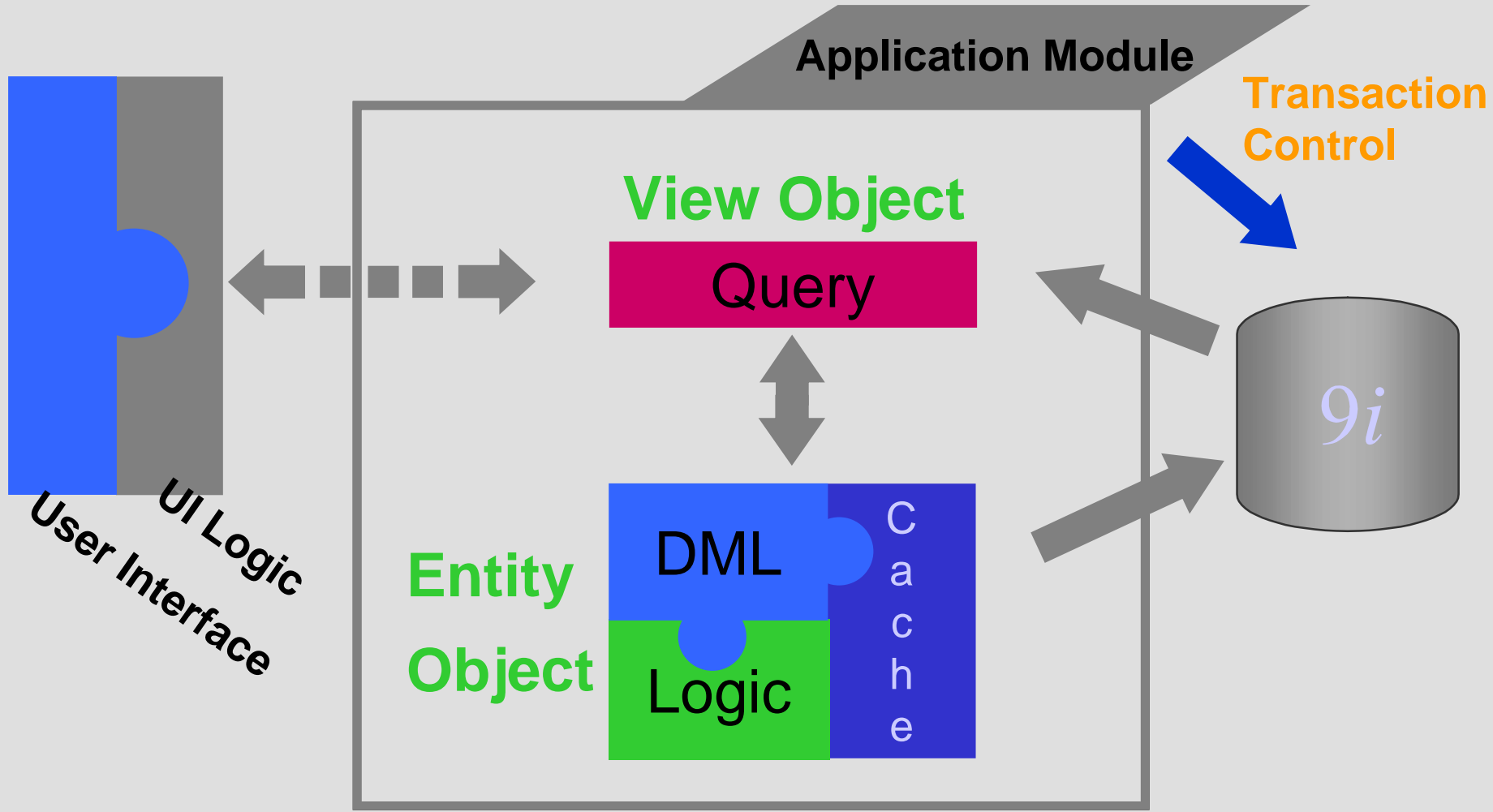


## **Application Modules:**

- ▲ **Zusammenfassung von VOs und View Links für die Unterstützung eines Anwendungsfalles**
- ▲ **Können weitere Dienste in Form von Custom Methods anbieten.**
- ▲ **Schnittstelle von BC4J Applikationen zum Client**
- ▲ **Bilden Komponenten, die auf unterschiedliche Plattformen verteilt werden können**
- ▲ **Definieren Transaktionskontext und Kontext für EO- und VO-Caches**



Quelle: Oracle



Quelle: Oracle

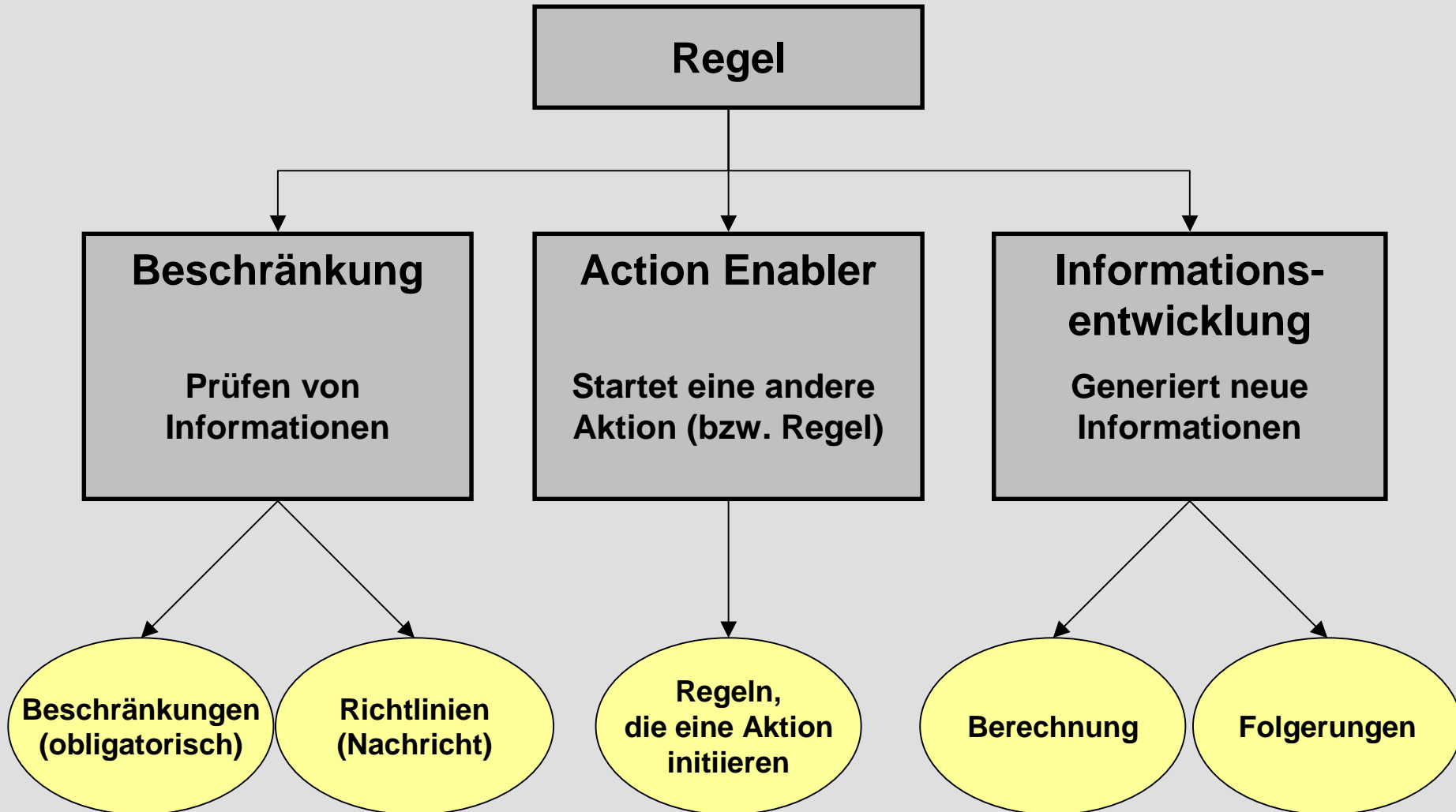


**Mit BC4J bietet Oracle ein Framework, das erprobte J2EE Design Patterns einfach benutzbar implementiert:**

- ▲ **Data Access Objects Pattern**
- ▲ **Value Object (Data Transfer Object) Pattern**
- ▲ **Fast-Lane Reader Pattern**
- ▲ **Page-by-Page Iterator Pattern**
- ▲ **Session Facade Pattern**

**Siehe „Simplifying J2EE and EJB Development with BC4J“  
von Steven Muench, Oracle Technet**

([http://otn.oracle.com/products/jdev/htdocs/j2ee\\_with\\_bc4j/j2ee\\_with\\_bc4j.html](http://otn.oracle.com/products/jdev/htdocs/j2ee_with_bc4j/j2ee_with_bc4j.html))





### ▲ Attribute

- Deklarativ:
  - Validators
  - Domains
- Java-Programmierung:
  - Set/get methods
  - Publish/Subscribe

### ▲ Entity Objects

- Lifecycle methods
  - create/remove
  - validate
  - before/after commit
  - postChanges/doDML
- Publish/Subscribe
- Custom methods

### ▲ View Objects

- Set/get methods
- Publish/Subscribe
- Custom methods

### ▲ Application Modules

- Custom methods



## Neues bei BC4J:

- ▲ Oracle9i JDeveloper unterstützt u.a. UML Klassendiagramme
- ▲ So können auch BC4J Modelle per Drag & Drop von DB Tabellen in einem Klassendiagramm erstellt werden
- ▲ Unterstützt das Erstellen von DB Tabellen aus BC4J Komponenten
- ▲ Synchronisation von BC4J Modellen und Datenbank
- ▲ Polymorphic Row Sets
- ▲ M-zu-N-Relationen
- ▲ . . .

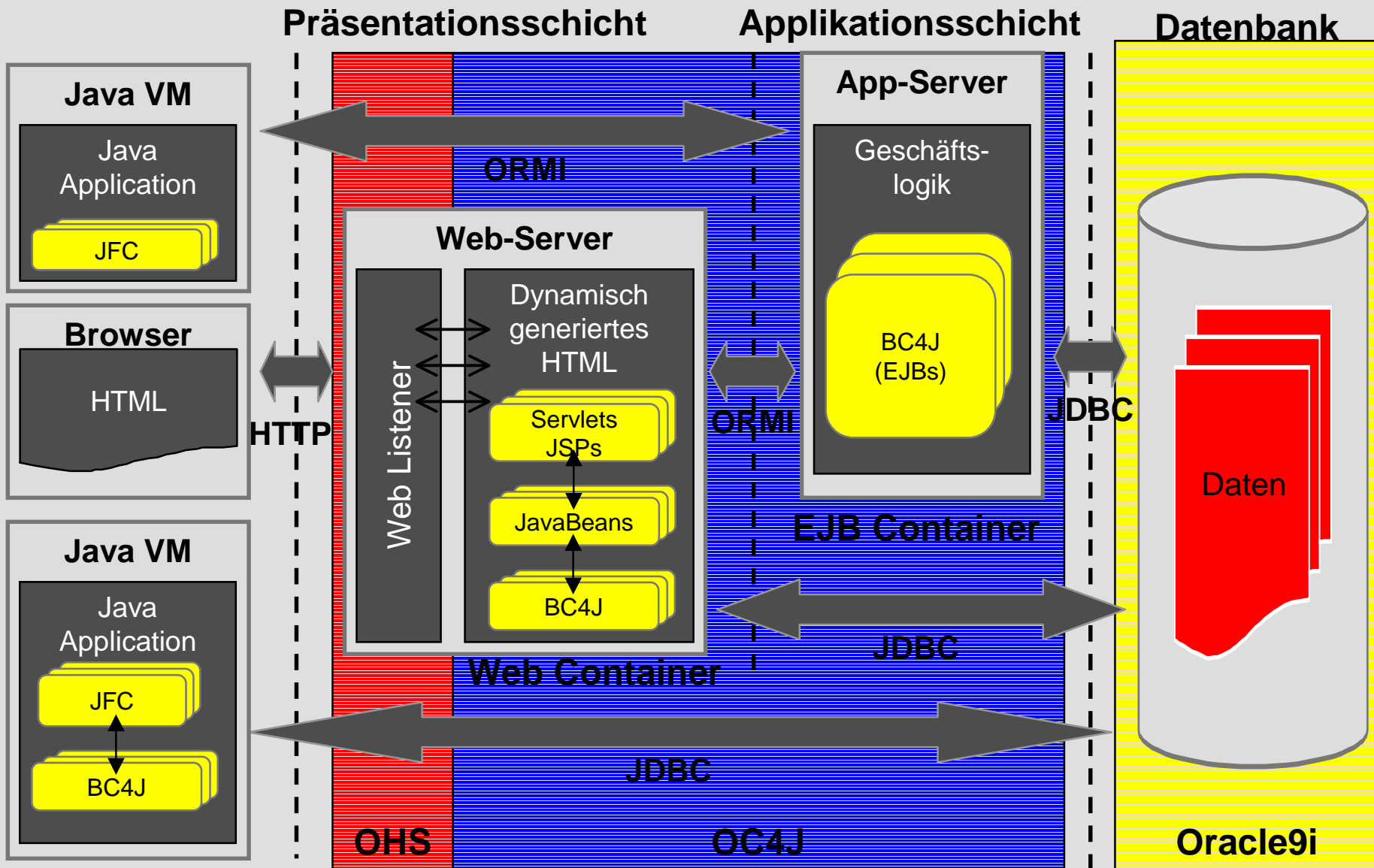


**Das BC4J Framework unterstützt eine flexible Verteilung auf unterschiedliche Plattformen ohne Änderungen der Anwendungskomponenten:**

- ▲ **Als Enterprise JavaBeans**
- ▲ **Als Java CORBA Komponenten**
- ▲ **„Lokales“ Deployment**

**Oracle9i JDeveloper unterstützt das Deployment nach**

- ▲ **Oracle9i Application Server (OC4J)**
- ▲ **BEA WebLogic**
- ▲ **Visibroker**





- ▲ **BC4J unterstützt effizient die Entwicklung von Datenbank-basierten Applikationen. Die Anwendungsentwicklung kann sich auf die fachlichen Anforderungen konzentrieren.**
- ▲ **BC4J bietet neben der Persistenzschicht auch umfangreiche Möglichkeiten zur Implementierung komplexer Geschäftslogik.**
- ▲ **BC4J Applikationen zeichnen sich durch saubere Schichtentrennung aus und lassen sich leicht auf unterschiedliche Plattformen verteilen.**
- ▲ **Für BC4J Applikationsmodule lassen sich leicht unterschiedliche Java Clients (HTML, UIX, JSP, Swing) entwickeln.**



# D e m o



**F & A**