

ORACLE

Hardware and Software

ORACLE®

Engineered to Work Together



ORACLE[®]

Oracle ADF – PL/SQL-Logik ersetzen oder wiederverwenden ?

Jürgen Menge

Oracle Deutschland B.V. & Co. KG

Ausgangssituation

PL/SQL in bestehenden Forms/Reports-Anwendungen

- Hauptteil der Geschäftslogik (PL/SQL) befindet sich in der Oracle-Datenbank
- Vorteil der einheitlichen Sprache in Client und Server

Was wird aus der Geschäftslogik (PL/SQL) bei einem Umstieg auf Java EE bzw. Oracle ADF ?



Was wird aus der PL/SQL-Logik ?

Ersetzen von PL/SQL durch Java in der Mittelschicht

- Re-Implementierung der Geschäftslogik
- Vorteile der Java-Plattform (Integration, Offenheit, Weiterentwicklung)
- Skalierbarkeit und Ausfallsicherheit der Middleware

Was wird aus der PL/SQL-Logik ?

Wiederverwenden des PL/SQL in ADF-Applikationen

- Bewahren des Investments
 - Neuentwicklungen können schneller umgesetzt werden
 - PL/SQL-Erfahrung in den Entwickler-Teams (Produktivität)
 - paralleler Betrieb von bestehenden (Forms-) und neuen (ADF-) Applikationen möglich
 - Vertrauen in die Beständigkeit von PL/SQL
- ⇒ Kapselung der PL/SQL-Logik, so dass ein späterer Austausch möglich wäre

Arten von Geschäftslogik

Tabellenzugriff und Berechnungen

- Datenzugriff über Table API bzw. Data Access Layer (DAL)
 - Datenzugriff erfolgt über eine spezielle Zugriffsschicht (Select, Insert, Update, Delete, Lock)
 - Implementierung der Zugriffsschicht meistens als Kombination aus Views für lesenden Zugriff und Packages für DML-Operationen
- Validierungen, Berechnungen, Ableitungen
 - Implementierung der Logik in PL/SQL-Prozeduren oder –Funktionen
- Zumeist in Kombination oder in aufeinander aufbauenden Schichten
 - Fachkomponenten ↔ Datenkomponenten

Herausforderungen bei der Umsetzung

Abbildung von Datentypen (Java ↔ PL/SQL)

- Verwendung von Standard-Java- oder Oracle-Klassen für die Java-Datentypen
 - Verwendung der Oracle-Klassen in reiner Oracle-Umgebung, da sie mehr Funktionalität und Performance bieten
- Einschränkungen in Java (JDBC)
 - keine Unterstützung für Record Types
 - keine Unterstützung von *%rowtype* und *table of %rowtype*
 - kein Datentyp Boolean in Java
- globale Package-Variablen sind bei Connection Pooling nicht verwendbar

Prinzipien der Umsetzung

Kapselung der PL/SQL-Logik

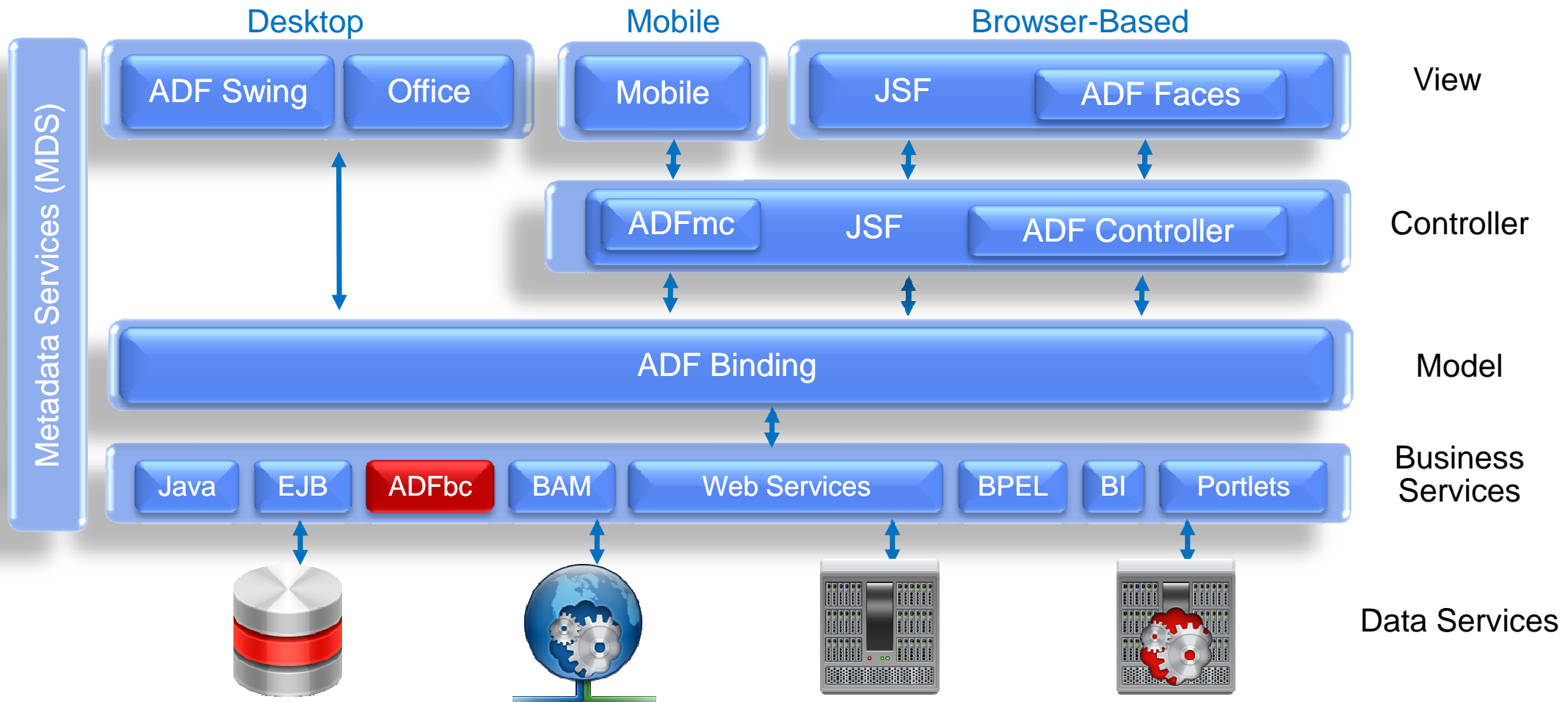
- Wrapper-Schicht für den Zugriff auf PL/SQL
 - PL/SQL Object Types zur Nachbildung von Record Types
 - Abbildung der PL/SQL Object Types auf Java-Klasse *oracle.sql.Struct*
 - ermöglicht späteren Austausch der PL/SQL-Logik
- Oracle JPublisher kann als Utility zur Generierung eingesetzt werden
 - verfügbar im JDeveloper und mit der Datenbank

Persistenz-Frameworks

Framework-Unterstützung für PL/SQL

- Persistenz-Frameworks (ADFbc, TopLink/EclipseLink/JPA) bieten Möglichkeiten zum Aufruf von PL/SQL in der Datenbank
- Frameworks wurden entwickelt, um direkt mit normalisierten Tabellenstrukturen zu arbeiten
- Datenzugriff über PL/SQL
 - wesentliche Funktionen der Frameworks können nicht genutzt werden
 - Funktionen der Frameworks müssen angepasst/überschrieben werden
 - Ausnahmen werden zur Regel, damit gehen die Vorteile der Frameworks verloren (Produktivität !)
Beispiel: Java Persistence API – Verwendung von nativen Queries anstelle von JPQL notwendig
- ADF Business Components bieten vergleichsweise noch die meisten Möglichkeiten, die Spezifika der Oracle-DB zu nutzen

Architektur – Oracle ADF



Nutzung der PL/SQL-Logik aus Oracle ADF

Datenzugriffe über Table API / Data Access Layer

- Variante 1
 - Lesender Zugriff über View Objects auf Basis von DB Views
 - Überschreiben der Methode `doDML()` für die Entity Objects (Klasse `EntityImpl`)

```
protected void doDML(int operation, TransactionEvent e) {  
    if (operation == DML_INSERT) {  
        super.insertUpdate(attributes, rowType, indType, "BEGIN events_pkg.ins(?, ?, TRUE); END;");  
    } else if (operation == DML_UPDATE) {  
        super.insertUpdate(attributes, rowType, indType, "BEGIN events_pkg.upd(?, ?, TRUE); END;");  
    } else if (operation == DML_DELETE) {  
        super.delete(pkAttributes, pkType, "BEGIN events_pkg.del(?,TRUE); END;");  
    } else // unlikely but you never know  
        super.doDML(operation, e);  
}
```

- betrifft nur die ADF-Anwendung
- <http://www.oracle.com/technetwork/articles/muir-designer-085584.html>



Nutzung der PL/SQL-Logik aus Oracle ADF

Datenzugriffe über Table API / Data Access Layer

- Variante 2
 - Lesender Zugriff über View Objects auf Basis von DB Views
 - Verwenden von INSTEAD-OF-Triggern in der Datenbank
 - Trigger zündet bei allen DML-Operationen auf der View

Nutzung der PL/SQL-Logik aus Oracle ADF

Datenzugriffe über Table API / Data Access Layer

- Variante 3

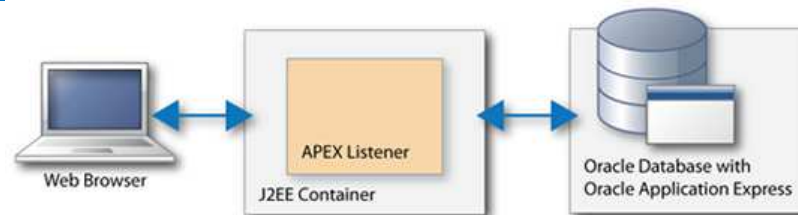
- Kapselung der PL/SQL-Logik als Web Service in der Datenbank
- Nativer SOAP Web Service

<http://sql-plsql-de.blogspot.co.uk/2012/05/soap-zugang-zu-plsql-database-native.html>

oder

- RESTful Web Service über den APEX Listener

http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058_01.pdf



- Nutzung des ADF Web Service Data Control oder Generierung eines Web Service Proxy im JDeveloper

Nutzung der PL/SQL-Logik aus Oracle ADF

Validierungen, Berechnungen, Ableitungen

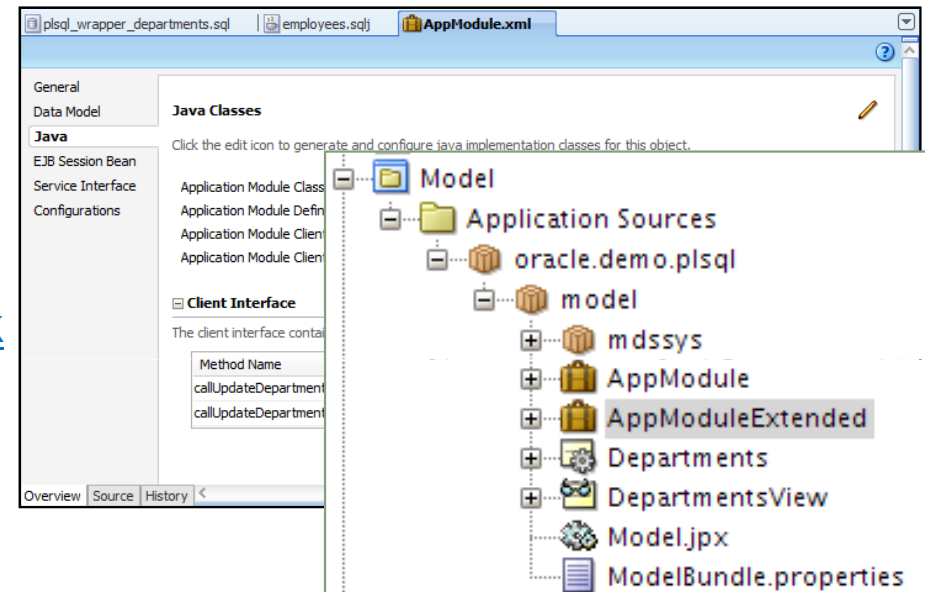
- Variante 1
 - Eigene Methoden für das Application Module schreiben (Klasse *AppModuleImpl*)
 - Aufruf der PL/SQL-Routinen über Callable Statements (Funktionen) oder Prepared Statements (Prozeduren)

```
// Method with prepared statement to call a PL/SQL procedure in the database
//
public void callUpdateDepartmentName (int deptNo, String newName)
{
    PreparedStatement plsqlBlock = null;
    String statement = "BEGIN update_dept_name_proc(:1,:2); END;";
    plsqlBlock = getDBTransaction().createPreparedStatement(statement,0);
    try
    {
        plsqlBlock.setInt(1,deptNo);
        plsqlBlock.setString(2,newName);
        plsqlBlock.execute();
    }
    catch (SQLException sqlException)
```

Nutzung der PL/SQL-Logik aus Oracle ADF

Validierungen, Berechnungen, Ableitungen

- Variante 1 (Fortsetzung)
 - Registrieren der Methoden als Client Interface
 - Tipp <http://www.youtube.com/user/ADFTalk>
 - Ableiten (Extend) des AppModule
 - Verwenden des abgeleiteten AppModule in der Applikation
 - Sichtbarkeit des AppModule ausschalten
 - D. Mills/P. Koletzke/A. Roy-Faderman, Oracle JDeveloper 11g Handbook: A Guide to Fusion Web Development (Oracle Press), 2009



Nutzung der PL/SQL-Logik aus Oracle ADF

Validierungen, Berechnungen, Ableitungen

- Variante 2

- Kapselung der PL/SQL-Logik als Web Service in der Datenbank

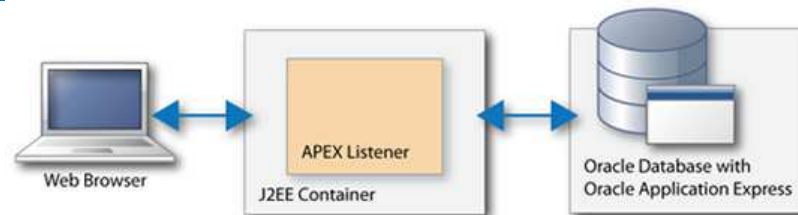
- Nativer SOAP Web Service

<http://sql-plsql-de.blogspot.co.uk/2012/05/soap-zugang-zu-plsql-database-native.html>

oder

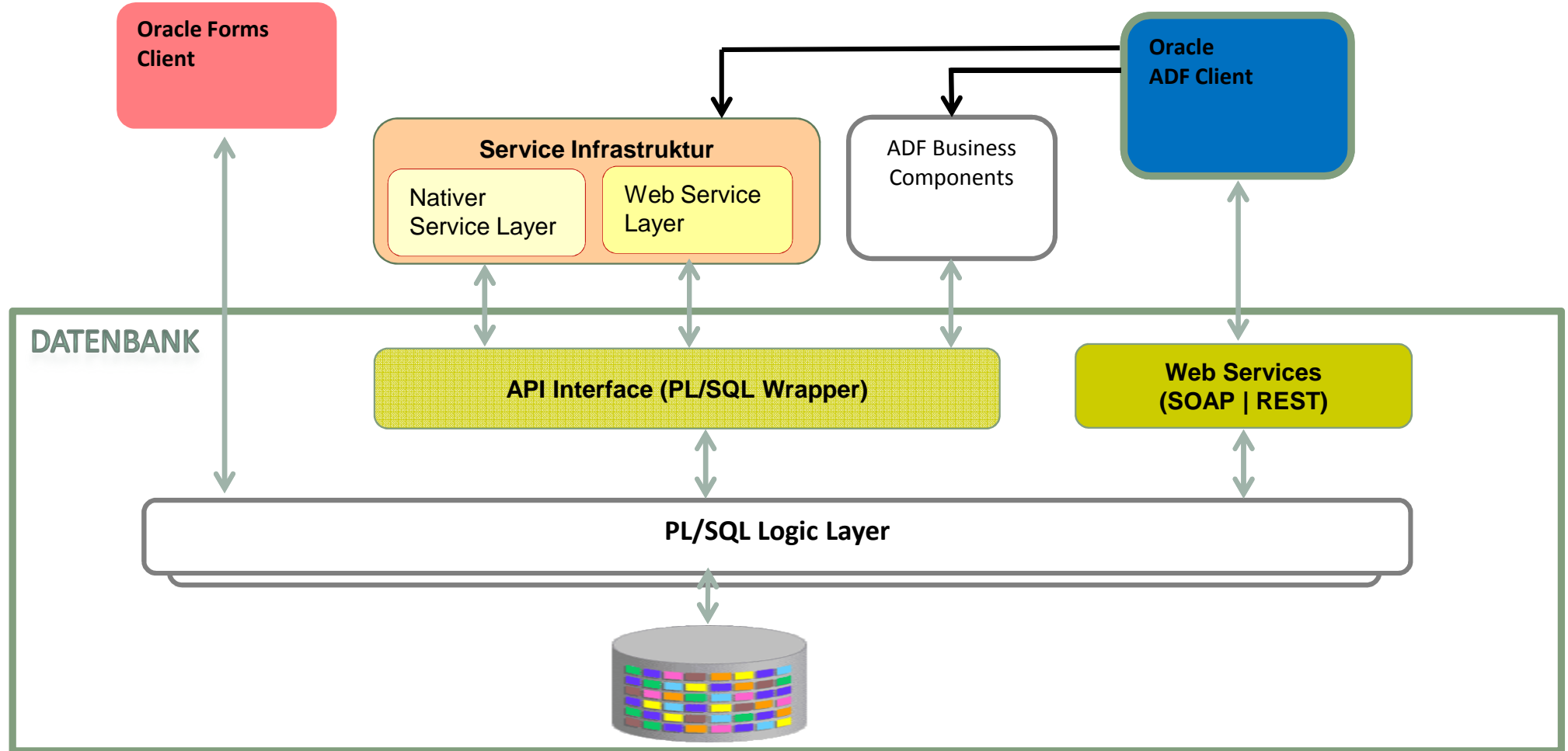
- RESTful Web Service über den APEX Listener

http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058_01.pdf



- Nutzung des ADF Web Service Data Control oder Generierung eines Web Service Proxy im JDeveloper

Mögliche Ziel-Architektur





Vorgehensweise

Änderungen an den Datenstrukturen

- Nachziehen von Änderungen an den Datenstrukturen über alle Schichten
- Generischer Ansatz
 - viele Fehler werden erst zur Laufzeit entdeckt
 - <http://mohammedsaadadf.blogspot.de/2012/05/executing-stored-procedure-and.html>
- Generativer Ansatz
 - automatisierte Generierung aller Komponenten bei Änderungen
 - <http://www.youtube.com/user/ADFTalk>

Wichtige Aspekte beim Aufruf von PL/SQL

- Aktualisierung der Caches in der Mittelschicht bei Änderungen, die durch die PL/SQL-Logik in der Datenbank ausgeführt werden
- Wo liegt die Kontrolle über die Transaktionen ?
- Authentifizierung und Autorisierung
 - Dedizierte Datenbank-Verbindungen vs. Connection Pooling
 - als Übergangslösung evtl. Proxy Authentication verwenden
 - kein Session State in der Datenbank
- Locking
 - Pessimistisches vs. optimistisches Locking

Oracle ADF Communities

- Google Group „ADF Methodology“
<http://groups.google.com/group/adf-methodology>



- Deutsche Oracle ADF Community

- Oracle ADF Community auf XING
- Twitter: ADFCommunityDE
- News Sessions (30-minütige Web Seminare an jedem 2. Freitag)
- regelmäßige Treffen der Oracle ADF-Partner
- Zusammenarbeit mit der DOAG (Jahreskonferenz, SIG)
- Hacker's Event



Fragen & Antworten



Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®