



# PL/SQL in ADF

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen

Hendrik Gossens

Nürnberg, 22.11.2012

# MT AG

## Daten und Fakten

Als anerkannter Partner der führenden Technologiehersteller verbinden wir die Agilität eines mittelständischen Unternehmens mit der Lösungskompetenz internationaler Beratungshäuser.

### Hauptsitz

Ratingen

### Niederlassungen

Hamburg, Dortmund, Frankfurt,  
Luxemburg

### Tochtergesellschaften

MT-ifs GmbH, MT-ics GmbH

### Beschäftigte (2012)

220 Festangestellte  
80 Freiberufler



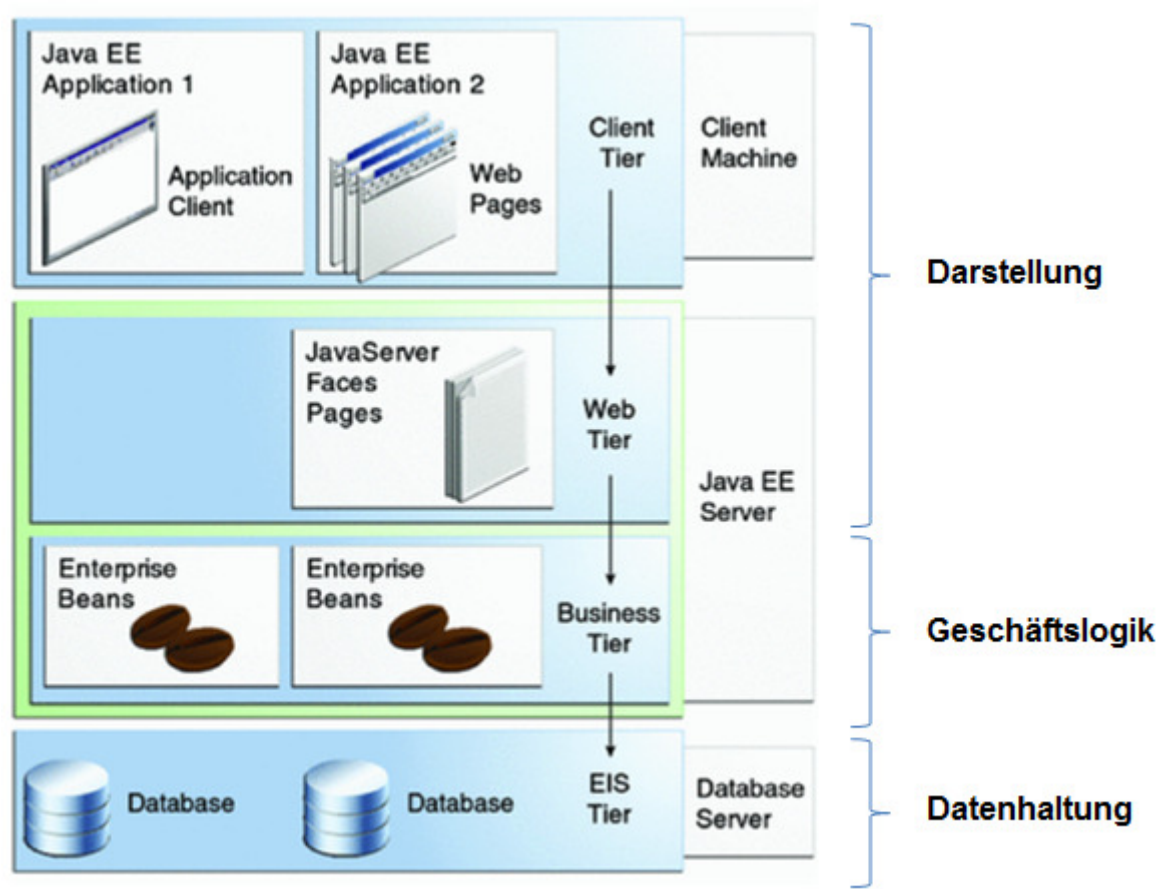
# Agenda

1. Problemstellung
2. Java EE Architektur
3. Möglichkeiten zur Wiederverwendung von PL/SQL
  - Business Components
  - Webservice
4. Show Cases
  - Business Components
  - Webservice
5. Fazit

# Problemstellung

- Klassische Forms-/Anwendungen im Unternehmen
- Geschäftslogik als PL/SQL Code in der Datenbank
- ADF ist Java EE Framework => Geschäftslogik als Java Code in der Mittelschicht
- Fragen:
  - Was geschieht mit etabliertem und getestetem PL/SQL Code?
  - Sind getätigte Investitionen bei Umstieg auf ADF verloren?

# Java EE Architektur

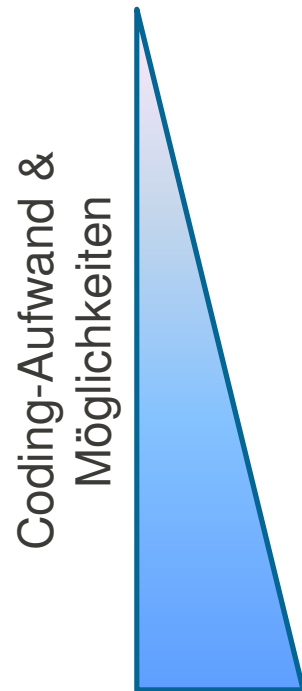


<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

# Wiederverwendung – aber wie?

- Architekturunterschiede zwischen Forms/Reports und ADF
- Bei Wiederverwendung Möglichkeit der späteren Transformation in die reine ADF Welt offen halten!
- Lösung: Virtuelle Geschäftslogik-Schicht für den PL/SQL-Code
  - => später durch Neuimplementierung austauschbar
  - => Schrittweise Migration in neue Welt möglich!

# Kapselung über Business Components



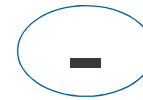
- Drei Methoden:
  1. Lesender Zugriff auf Datenbank Views via BC und INSTEAD OF-Trigger für DML
  2. Lesender Zugriff auf Datenbank Views via BC, Methode `doDML()` in Entity Object Implementierungsklasse überschreiben.
  3. Eigene Methoden in der Implementierungsklasse des Application Module schreiben
    - Nativer Zugriff auf PL/SQL via JDBC
      - Für PL/SQL Prozeduren: `PreparedStatement`
      - Für PL/SQL Funktionen: `CallableStatement`

# Bewertung

Business Components und INSTEAD OF Trigger



- Kein Anpassungsaufwand in Business Component Klassen



- Zusätzliche Logik in Datenbank

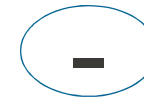


# Bewertung

doDML () in Entity Object Implementierungsklasse überschreiben



- Easy to use
- Geringer Anpassungsaufwand in Business Component Klassen



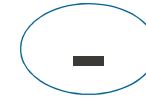
- Unterscheidbare Fälle: Insert, Update, Delete
- Somit nur für kleinere Anwendungsfälle geeignet

# Bewertung

Eigene Methoden in AM-Implementierungsklasse



- Für Java-Entwickler einfach umsetzbar
- Kein zusätzlicher Code in der Datenbank



- Nicht deklarativ
- Deklarativer Ansatz muss bei Neuimplementierung/Schrittweise Migration nachgezogen werden  
=> Erhöhter Aufwand

# Kapselung als Webservice

- Zwei Möglichkeiten
  1. SOAP-Zugang zu PL/SQL via **Database Native WebService**
  2. Als RESTful Service via **APEX Listener**
  
- Services dann jeweils entweder über das Webservice-**DataControl** oder über Webservice-**Proxy-Klasse** in ADF einbinden

# Database Native WebService

- Benötigt XML DB in der Datenbank
- User muss Berechtigungen `xdb_webservices` und `xdb_webservices_over_http` haben
- Database Native WebServices müssen einmalig in der XML DB aktiviert werden
- HTTP-Protokollserver muss aktiv sein
- **WSDL:** `http://{host}:{port}/{servlet}/{USER}/{FUNCTION}?wsdl`

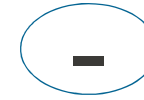
(<http://sql-plsql-de.blogspot.de/2012/05/soap-zugang-zu-plsql-database-native.html>)

# Bewertung

## Database Native Webservice



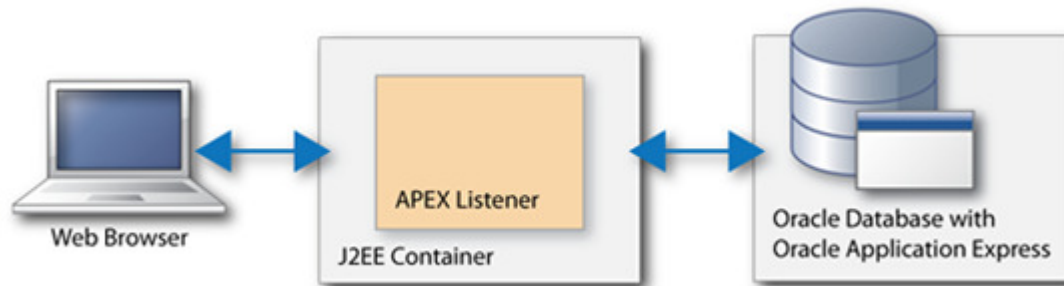
- deklarativ in ADF einbindbar
- Lose Kopplung
- Gute Unterstützung für schrittweise Migration



- Konfigurationsaufwand Datenbank (XML DB)

# RESTful Service via APEX Listener

- Für Produktivumgebungen: Release 1.1.4 - July 2012
- Early Adopter: APEX Listener 2.0



(<http://www.oracle.com/technetwork/developer-tools/apex-listener/overview/index.html>)

# RESTful Service via APEX Listener

- 4 Modi
  1. Standalone
  2. WebLogic
  3. GlassFish
  4. Oracle Containers for J2EE (OC4J)

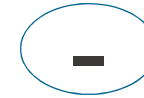
([http://docs.oracle.com/cd/E21611\\_01/doc.11/e21058/install.htm](http://docs.oracle.com/cd/E21611_01/doc.11/e21058/install.htm))

# Bewertung

## RESTful Service via APEX Listener



- REST aktueller Trend
- deklarativ in ADF einbindbar
- Lose Kopplung
- Gute Unterstützung für schrittweise Migration
- APEX Listener in Application Server lauffähig => Zielarchitektur
- Einfache Integration mit JAVA-basierter Middleware



- Konfigurationsaufwand für Listener und Services





# Show Cases

# PL/SQL via Webservice - Do it yourself

- „Database App Development VM“ unter <http://www.oracle.com/technetwork/community/developer-vm/index.html>
- Netzwerkadapter hinzufügen und NAT Port Forwarding für Ports 1521 und 8888 festlegen
- APEX Listener gemäß Anleitung ([http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058\\_01.pdf](http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058_01.pdf)) lokal installieren.
- hr-Schema in Datenbank freischalten (`sys as sysdba/oracle`)
- Passwort für APEX\_PUBLIC\_USER: `oracle`

# Fazit

- Unterschiedliche Architekturparadigmen zwischen Forms/Report und Java EE / ADF
- Bestehender PL/SQL Code kann trotzdem wiederverwendet werden
- Parallelbetrieb klassischer Forms/Reports und ADF Anwendungen auf der selben Code-Basis
- Zielarchitektur bleibt trotzdem die Java EE Architektur (Schichtentrennung)
- Schrittweise Migration durch Kapselung

# Quellen

- Vortrag DOAG Development 2012: Jürgen Menge - Oracle ADF - bestehende PL/SQL-Logik ersetzen oder wiederverwenden?
- <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- <http://www.oracle.com/technetwork/articles/muir-designer-085584.html>
- <http://www.youtube.com/watch?v=gOUUIIRjDpU&feature=plcp>
- [http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058\\_01.pdf](http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058_01.pdf)
- <http://sql-plsql-de.blogspot.de/2012/05/soap-zugang-zu-plsql-database-native.html>
- <http://www.oracle.com/technetwork/articles/muir-designer-085584.html>
- <http://www.oracle.com/technetwork/developer-tools/jdev/index-092937.html>

# Besuchen Sie auch unsere weiteren Vorträge auf der DOAG 2012

Dienstag, 12 Uhr, Raum Riga

Dienstag, 13 Uhr, Raum Seoul

Dienstag, 14 Uhr, Raum Stockholm

Dienstag, 15 Uhr, Raum Kopenhagen

Dienstag, 16 Uhr, Raum Stockholm

Mittwoch, 13 Uhr, Raum Riga

Mittwoch, 15 Uhr, Raum Riga

Mittwoch, 16 Uhr, Raum Seoul

Donnerstag, 09 Uhr, Raum Istanbul

Donnerstag, 14 Uhr, Raum Konf. EG

Donnerstag, 15 Uhr, Raum Istanbul

Donnerstag, 16 Uhr, Raum Oslo

Dynamisch Unterschiede in Datensätzen auf Feldebene finden by S.O. Kelbert

Route to ASM by Ernst Leber

Automatische Generierung der ETL-Prozesse OWB vs. ODI by Irina Gotlibovych

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen by Hendrik Gossens

„Managed Code“ mit OWB – Methoden und Wege by Bernhard Rosenberger

Dateizugriff mit new I/O 2 by Wolfgang Nast

WebServices in Java SE und EE by Wolfgang Nast

Das Mysterium OPatch by Volker Mach

Das größte APEX Projekt der Welt @ Union Investment by Niels de Bruijn

Testen mit Pfefferminzgeschmack by Birgit Kratz

APEX goes UNIT Testing by Oliver Lemm

SOA verspielt – rekursive BPEL Prozesse by Guido Neander





# Vielen Dank.

**MT AG**

Balcke-Dürr-Allee 9  
40882 Ratingen

Telefon: +49 (0) 21 02 309 61-0  
Telefax: +49 (0) 21 02 309 61-10

E-Mail: [info@mt-ag.com](mailto:info@mt-ag.com)  
[www.mt-ag.com](http://www.mt-ag.com)