



"Write Once, Run Nowhere (else)?"

Portabilität von J2EE-Anwendungen

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Tobias Kieninger

Gliederung

- **Motivation**
- Entwicklungsszenario
- Typische Verdächtige

Was wir (nicht) tun!



- Ist Websphere performanter als BEA?
- Skaliert Orion besser als Jboss?
- Hat Resin mehr Features als Tomcat?
- ~~Welcher Application Server hat die besten Mapping Tools?~~

3

Ziele der Session!



- Mögliche Lösungsansätze
- Richtlinien
- Mögliche Entwicklungsprozesse
- Mögliche Integrationsprozesse

4

Portabilität! Warum?



- Wir haben doch einen Standard ;-)
- Was steht leider nicht drin?
 - Persistenzmapping
 - Administration
 - **JMX teil von J2SE**
 - Deployment
 - **Deployment API ab J2EE 1.5**

5

Portabilität! Warum?



- Unabhängigkeit vom Lieferanten eröffnet Verhandlungsspielräume
- Verkauf / Weitergabe des Produkts - verschiedene Zielplattformen geplant
- Entwicklungsbeginn vor Serverentscheidung
- Serverentscheidung muss nachträglich geändert werden
 - politische Gründe (Stichwort „immer IBM“)
 - ökonomische Gründe (Stichwort „JBoss ist doch umsonst...“)
 - technologische Gründe (Stichwort J2EE 1.4 und BEA)

6

Gliederung



- Motivation
- **Entwicklungsszenario**
- Typische Verdächtige

7

Projektkennzahlen



- 8 Leute
- ca. 3,5 Personenjahre
- ca. 60 Entitäten
- ca. 220 Formulare
- Gesamtlaufzeit ca. 1,5 Jahre

8

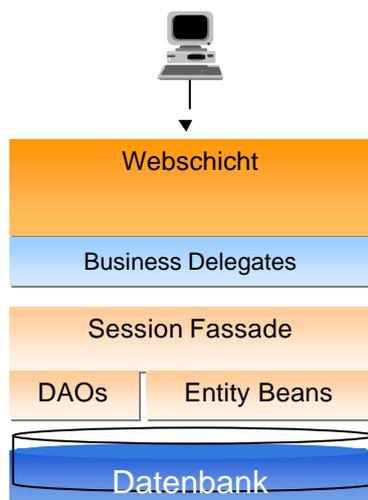
Voraussetzungen zum Projektbeginn



- **Kundenwunsch**
 - **Zielsystem:** Websphere, Oracle
- **Entwicklerwunsch**
 - **Entwicklersysteme:** JBoss, Eclipse, Open Source

9

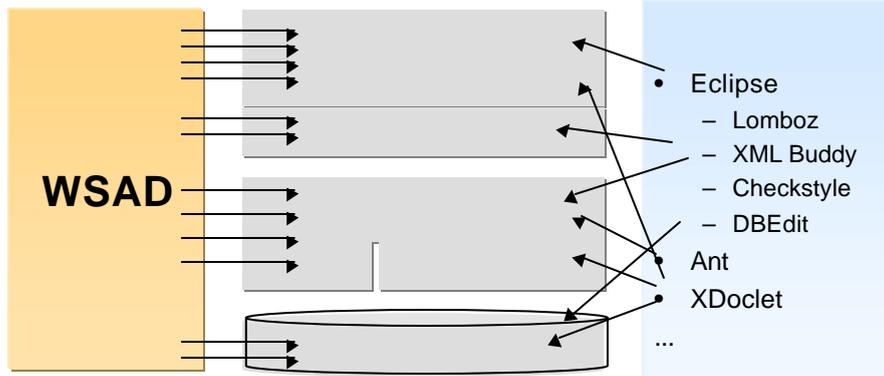
Grobe Architektur



- **Frontend**
 - Servlets / JSP
 - Struts
- **Backend**
 - EJBs
 - J2EE-Patterns
 - DAOs
- **Datenbank**
 - Relational

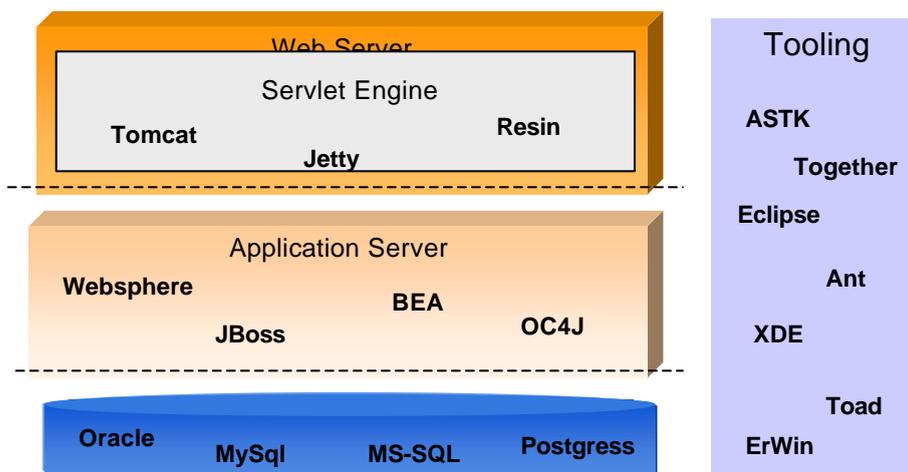
10

Tooling, ein wichtiges Thema!



11

Server, Tools, Datenbanken



12

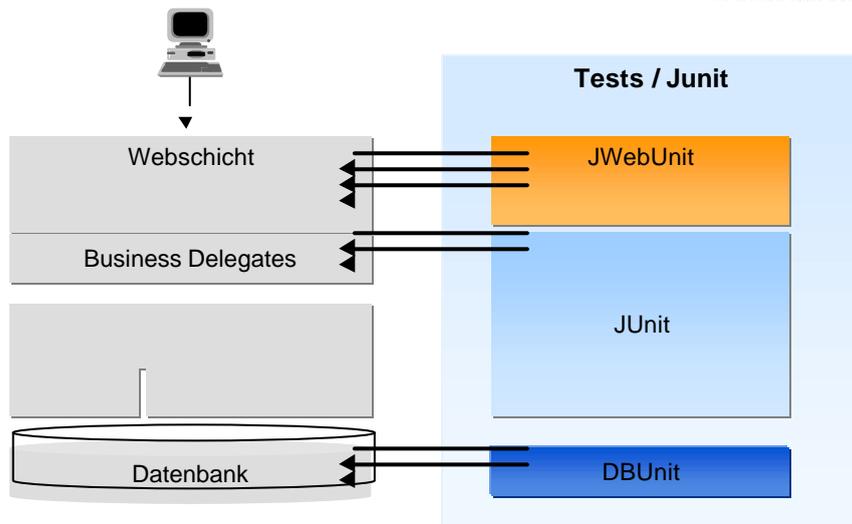
XP - the central principles



- Fast and detailed feedback
 - test driven design with acceptance tests
 - Customer On-site
 - pair programming
- Common understanding
 - planning game
 - simple design
 - System metaphor
 - collective code ownership
 - coding conventions
- Continuous process:
 - continuous integration
 - refactoring
 - frequent and small releases
- Developer Welfare
 - a week has 40 hours

13

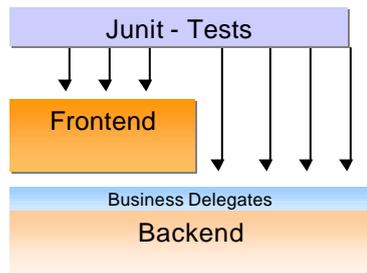
Testing



14

JUnit

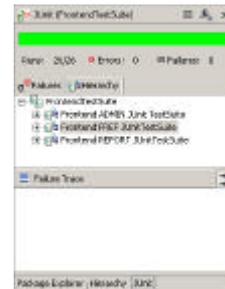
- Basis-Treiber für zahllose Derivate und damit einheitliche Basis für autom. Tests
- Unit-Tests für Business Delegates



15

JWebUnit

- Basiert auf HTTPUnit
- Simuliert Webbrowser
- Prüft Ergebnisseiten ab



```
...  
public void testCreateAndDisplay() {  
    login("/admin/mandator/create.do");  
    assertEqualsKey("admin.mandator.create.title");  
    setFormElement("name", "Norbert Leitwein");  
    submit(BUTTON_SAVE);  
    assertEqualsKey("admin.mandator.create.title");  
    assertTextPresent("Sie müssen Feld Name ausfüllen.");  
}
```

16

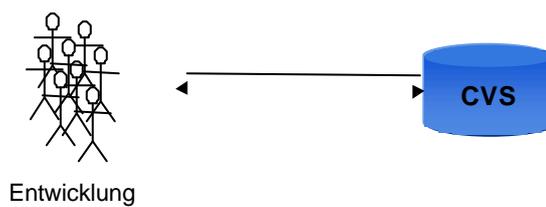
DBUnit

- Schreibt Testdaten in Datenbank
- Datenbank wird für **jeden** Test neu initialisiert

```
...  
<ID_TABLE ENTITY="de.project.AccountBean" LASTID="1"/>  
<ACCOUNT ID="1" NAME="Diba Kto" NUMBER="0815" ... />  
<ARTEFACT ID="1" NAME="Gegenstand" NOTE="Bemerkung" .../>  
<ARTEFACT ID="2" NAME="Gegenst 2" NOTE="Bemerkung2" .../>  
...
```

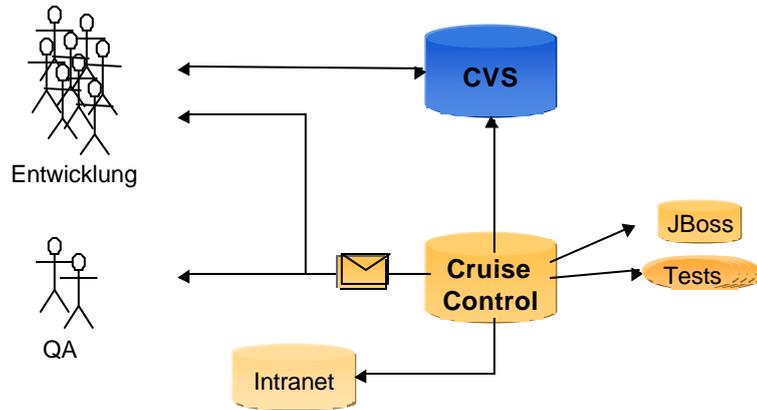
17

CVS - Concurrent Versions System



18

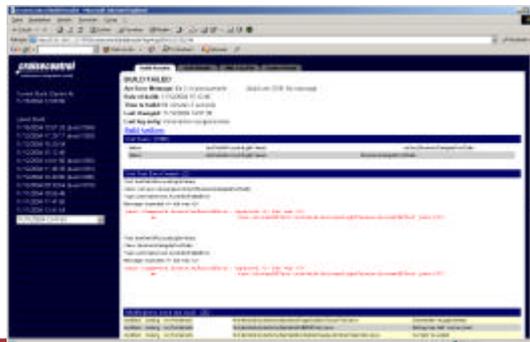
A build a day... Continuous Integration!



19

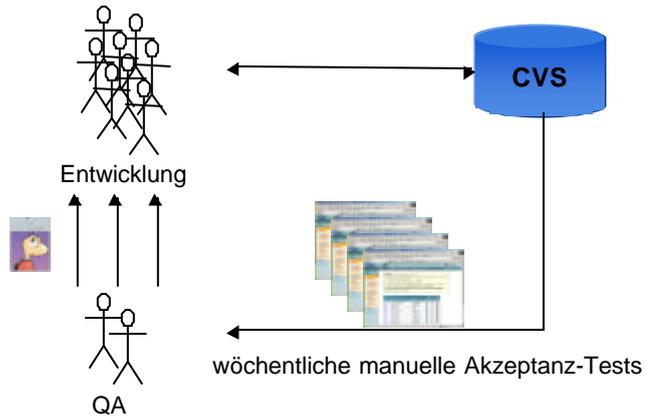
Cruise Control

- Ist im Prinzip ein Timer
- der Ant-Skripte anstoßen
 - und Mails verschicken
 - und Reports ablegen
- kann



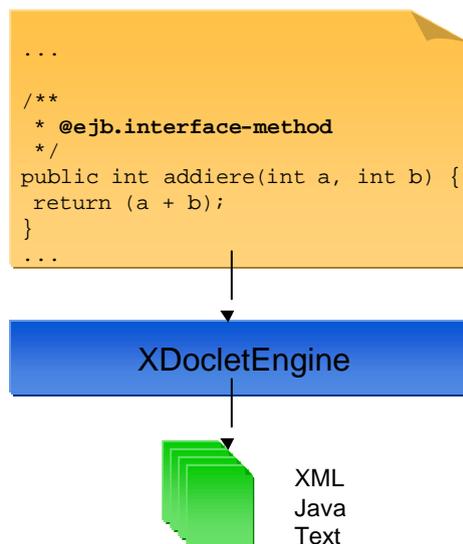
20

Wöchentliche Milestonebuilds



21

XDoclet



22

XDoclet im Projekt



- **EJB**
 - SessionBeans
 - EntityBeans
 - **CMR Relationen**
 - **Mappinginformationen**
 - Value Objects / Transfer Objects
 - Session Fassade
 - Struts Formulare
 - **Mit Validierung**
- **Web**
 - Servlets
 - Listener
 - Struts
 - **Actions**
 - **ActionForms**

23

XDoclet Beispiel



```
/**
 * @ejb.bean description="value" display-name="Account"
 * local-jndi-name="local/AccountLocalHome"
 * name="Account" primkey-field="id" schema="Account"
 * type="CMP" cmp-version="2.x" view-type="local"
 *
 * @ejb.pk class = "java.lang.Long"
 * ...
 *
 * @ejb.value-object name = "Account" match = "Account"
 * ...
 * @struts.form name = "Account"
 * ...
 */
public abstract class AccountBean implements EntityBean {
    ...
}
```

24

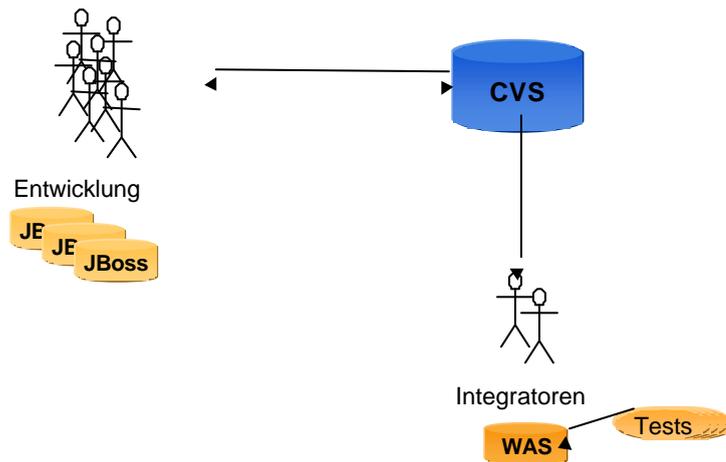
XDoclet und Portabilität



- JBoss-Tags
 - Bsp: @jboss.query
 - mehr als 30 Tags
 - Gut gepflegt
- Websphere-Tags
 - DB Mapping
 - Schwer verwendbar

25

Strategie: Seltene aber regelmäßige Portierung



26

Regelmäßige, seltene Portierung



- Vorteile
 - Bewahrung der Arbeitsweise
 - Kenntnisse für verschiedene Server können auf wenig Leute reduziert werden
- Nachteile
 - Zusätzliche Migrationsaufwände
 - **Aufwand bei ca. 2 BT pro Release**
 - **Risiko minimal durch automatisiertes Testbett**
 - Serverspezifische Probleme des Zielsystems werden erst bei der Portierung entdeckt

27

Fazit

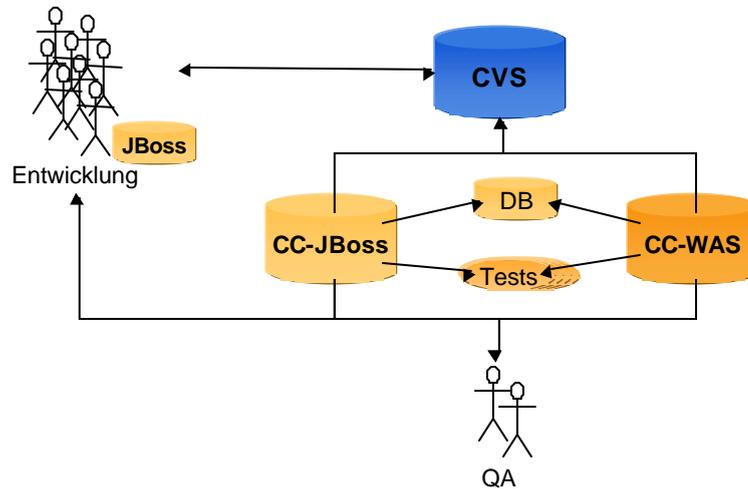


- Migrationsaufwand zufriedenstellend
- Prozess verbesserungsfähig

- Team glücklich
- Kunde glücklich

28

Aktuelle Erweiterung



29

Automatische Integration und Tests auf mehreren Servern

- Vorteile
 - Frühe Fehlererkennung
- Nachteile
 - Hoher Wartungsaufwand
 - **Automatisches Deployment schwierig in Websphere**
 - Mehr Kenntnisse im Team erforderlich

30

Gliederung



- Motivation
 - Entwicklungsszenario
 - **Typische Verdächtige**
- **JDK**
 - Deployment
 - JAAS
 - EJBQL / Persistenz
 - „Stolpersteinchen“

31

JDK Versionen



	Websphere	JBoss
IBM JDK 1.3	bis WS 5.0	
IBM JDK 1.4	ab WS 5.1	
Sun JDK 1.3	 	✓
Sun JDK 1.4	 	✓

32

Gliederung



- Motivation
 - Entwicklungsszenario
 - **Typische Verdächtige**
- JDK
 - **Deployment**
 - JAAS
 - EJBQL / Persistenz
 - „Stolpersteinchen“

33

JBoss



- HotDeploy durch dynamic Proxies
- Fehlertolerant bei Referenzen

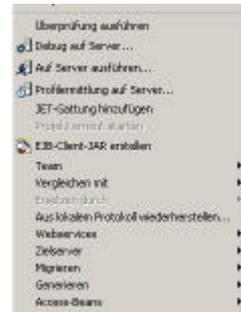


34

WSAD



- Für Stubs muss RMIC-Code generiert werden
- Kätzerei: „Deployen von 3 Beans dauert 20 Minuten“
- Streit mit den ANT-Tasks
 - Mitgeliefertes ANT verwenden
 - Fixpacks installieren



35

Gliederung



- Motivation
- Entwicklungsszenario
- **Typische Verdächtige**
 - JDK
 - Deployment
 - **JAAS**
 - EJBQL / Persistenz
 - „Stolpersteinchen“

36

JAAS



- Java Authentication and Authorization Service (JAAS)
 - seit Version 1.4
 - Java 2 Standard Edition
- Standardisierte Schnittstelle um einen Benutzer
 - anzumelden
 - autorisieren

37

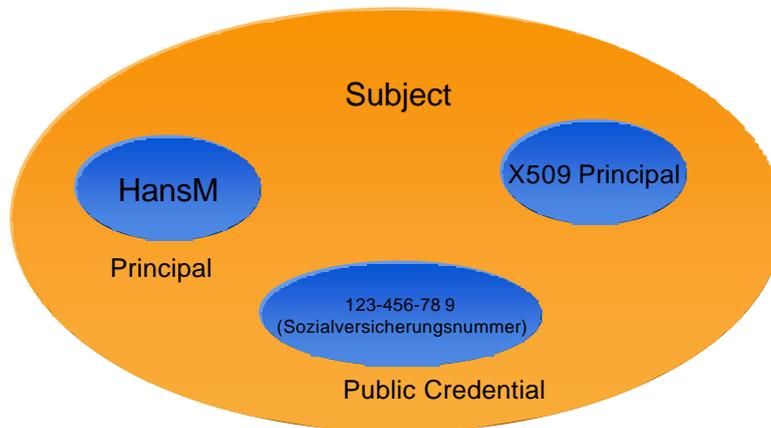
Deklaratives Modell



- Berechtigungen werden in XML Dateien beschrieben
 - ejb-jar.xml für EJBs
 - web.xml für Webanwendungen
- Gruppierung von Berechtigungen über Rollen
- keine Einbindung in Business Logik
 - weiterführendes kann integriert werden
- Trennung von Sicherheits- und Anwendungslogik

38

Subject und Principal



39

JAAS Grenzen

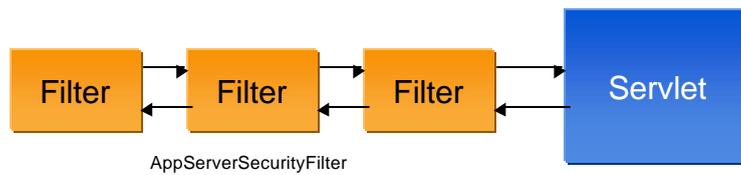
- Benutzername/ Passwort-Paar reicht nicht aus
 - z. B. zusätzliche Mandanteninformationen
- Schnittstelle zum Principal recht „schmal“
- Gibt `java.security.Principal` zurück
 - Interface mit 4 Methoden
 - `equals(Object o)`
 - `getName()`
 - `hashCode`
 - `toString()`



40

Zusätzliche Daten mit JAAS

- Bei Anmeldung werden zusätzliche Daten mit dem Benutzer verknüpft
 - Porträtiere Schnittstelle der Application Server
 - **WSSubject**
 - **ThreadLocal** bei JBoss



- Daten können später wieder ausgelesen und genutzt werden

41

CodeBeispiel

```
...
Long mandatorFK = RuntimeMandatorUtil.getMandator(sessionContext);
...

/**
 * Gibt die Mandanteninformation des angemeldeten Benutzers zurück
 * @param sessionContext Kontext
 * @return Id des Mandaten
 */
public static synchronized Long getMandator(SessionContext sessionContext) {

    if (ServerChecker.isJBoss()) {
        return MandatorAssociation.getMandator();
    }

    return getMandatorForWebSphere();
}
```

42

Gliederung



- Motivation
- Entwicklungsszenario
- **Typische Verdächtige**
 - JDK
 - Deployment
 - JAAS
 - **EJBQL / Persistenz**
 - „Stolpersteinchen“

43

Abfragesprache EJB QL



- nicht mehr Produktspezifisch
 - werden in der ejb-jar.xml Datei deklariert!
 - verwenden EJB-QL Abfragesprache

44

Beispiele für EJB QL



Einfache Abfrage

```
SELECT OBJECT(b)
FROM Bestellung AS b
```

Abfrage mit Parameter (aus Query oder Select)

```
SELECT OBJECT(b)
FROM Bestellung AS b
WHERE b.kundennr = ?1
```

Abfrage über Relationen hinweg:

```
SELECT OBJECT(b)
FROM Bestellung AS b, IN( b.positionen ) AS position
WHERE b.datum < ?1 AND
      position.produkt.name = 'Banane'
```

45

EJB-QL in der Praxis



- SQL nicht vollständig abgebildet, es fehlen z.B.
 - ORDER BY
 - GROUP BY
 - Subqueries
- Benutzen mehrerer Tabellen in einer Abfrage (Join)
 - in WebSphere nicht möglich
 - **SELECT OBJECT(o) FROM organisation AS o, Dossier as d**
WHERE d.applicantOrganisation.id = o.id and d.id = ?1
 - Spezifikation schwammig

46

EJB-QL in der Praxis II



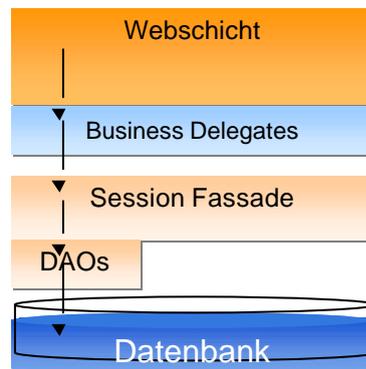
- Keine Beschränkung der Rückgabemenge
 - nur über Containererweiterungen
- Vergleichsoperatoren unzureichend
 - LIKE wird unterstützt, aber keine doppelten Wildcards
 - **Kri%ti%n nicht möglich**

47

Noch ein J2EE Pattern: Keine Entity Beans



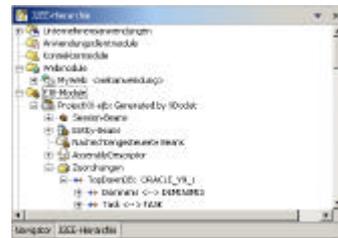
- Lesende Zugriffe teilweise als DAOs realisiert
 - Viele Joins
 - Komplexe Suchen
 - Grosse Ergebnismengen



48

DB Mapping

- Automatisches Mapping
 - Bei „grüner Wiese“ noch möglich
 - Aber wer hat das schon?
- WSAD
 - Mapping mittels Drag & Drop
- XDoclet
 - Als Tags



49

Gliederung

- Motivation
 - Entwicklungsszenario
 - **Typische Verdächtige**
- JDK
 - Deployment
 - JAAS
 - EJBQL / Persistenz
 - **„Stolpersteinchen“**

50

WEB-INF-Konfusionen



- **Best Practice:** JSP-Dateien vor direktem Zugriff schützen!
- **Wie?**
 - JSP-Dateien werden in /WEB-INF abgelegt
- **Konfusion?**
 - **Tomcat:** Dateien aus /WEB-INF dürfen gelesen werden
 - **SAP:** Kann Dateien dort **nicht** lesen

51

Out of Memory? WAS 5.0



- WAR-File grösser als 150 MB hat zu Out of Memory geführt
- Dateien nachträglich unterschieben als Workaround
- Fixpack installieren

52

Fazit



- Portabilität hat ihren Preis
- Portabilität liegt **selten** am Produkt, sondern an
 - Prozess
 - Spezifikationskonformität
- Portabilität ist realisierbar
 - Wichtig ist die Frage wie portabel man sein möchte

53

„Write Once, Run Nowhere?“ - Portabilität von J2EE-Anwendungen

© 2004 Orientation in Objects GmbH



Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de