

Geronimo Application Server die Alternative ist einsatzbereit



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Wer steht vor Ihnen?

- 10 Jahre Erfahrung in der Softwareentwicklung
 - Java EE Technologien
 - Java EE Application Server
 - Service Orientierte Architekturen
- Veröffentlichungen
 - Buch „Apache Geronimo im Einsatz“ als Autor
 - Mehrere Fachartikel
 - **JavaMagazin, XML & WebServices Magazin**
 - Präsentationen auf Konferenzen
 - **OOPSLA, JAX**
 - dedek@oio.de



2

Java und XML

) Akademie)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme

) Beratung)

- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

) Projekte)

- Schlüssel fertige Realisierung von Software
- Unterstützung laufender Projekte
- Pilot- und Migrationsprojekte

3

Agenda

- Einführung
 - Motivation (J2EE bis heute)
 - Problem
 - Geronimo
- Szenarien
 - Konfiguration- und Installationsoptionen
 - Mandantenfähigkeit



4

Java Enterprise Hype



- J2EE 1.x
 - standardisierte plattformunabhängige Betriebsinfrastruktur für die „neue Java Welt“
 - **multiusersicher**
 - **skalierbar**
 - **transaktional**
 - Entwicklungs- und Strukturierungsparadigma mit feingranularen Komponentenmodellen
- J2EE 5
 - serverunabhängiges Deployment
 - Verbesserung von Administration und Management der Anwendungen
 - „Ease of Development“

5

Newer Hypes ?



- Komponentenorientierung verliert ihre Bedeutung
 - CBD vs. SOA
 - sind EJB-Komponenten zu feingranular ?
 - stürzt J2EE ab wie CORBA?
 - „The recent release of Java EE 5 (formerly known as J2EE 1.5) has sparked a debate about the future of this popular platform. Although some industry watchers predict a rapid decline due to its growing complexity, its viability is rock solid.“ Gartner Group 9/2006
- leichtgewichtige Frameworks für POJO-Komponenten stellen zwar standardisiertes aber starres Modell von J(2)EE in Frage
 - Picocontainer
 - Spring
 - **“Strong Performer in Open Source Projects“ Q2/2006 Forrester**
 - Google Guice

6

Agenda



- Einführung
 - Motivation (J2EE bis heute)
 - **Problem**
 - Geronimo
- Szenarien
 - Konfiguration- und Installationsoptionen
 - Mandantenfähigkeit



7

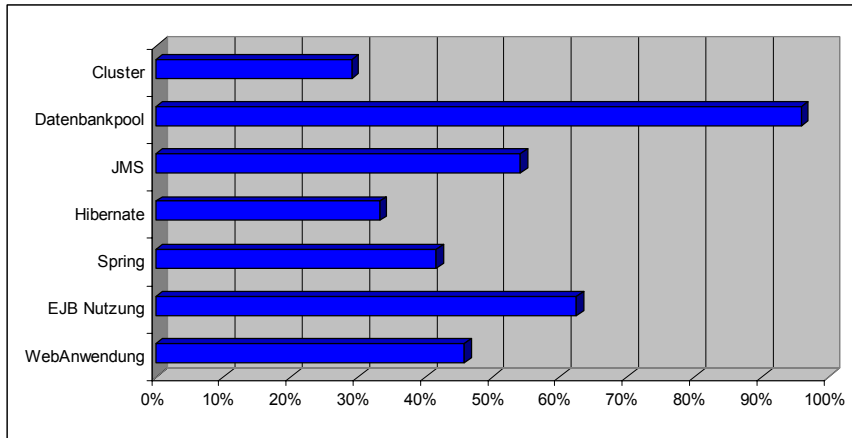
Java auf dem Server - aus OIO Kundensicht



- ca. 60% der Kunden nutzen Enterprise Java Beans
 - EJB Anwendungen entstanden vor 2-6 Jahren
 - in manchen Fällen EJB nur für Remoting
- 0% der Neuimplementierungen starten momentan mit EJBs
 - Spring/ Hibernate ist absoluter Hype
 - EJB 3/ JPA läuft langsam an
- Web Anwendungen im weiten Sinne überwiegen
 - HTTP Übertragung
 - nur manchmal reine RMI Übertragung

8

Bestandteile der Server Anwendungen - OIO Kundensicht (2001-2006)



9

Drei Lösungswege

- Full Feature J2EE-Server
 - Weblogic, WebSphere, Netweaver..
 - Geronimo, JBoss, Jonas, Glassfish..
- Kein Applicationsserver - Pure Spring
- Tomcat mit Erweiterungen

10

Full Feature JavaEE Server



- Pro
 - alle APIs unterstützt
 - **JMS, JTA, JCA...**
 - zertifizierte Kombination von Enterprise Services Komponenten in einer Serverdistribution
 - QoS-Angebote von Appserver-Anbietern
 - definierte Administrationsumgebung
 - Management und Monitoring „aus einer Hand“
- Contra
 - schwergewichtige Infrastruktur verbraucht Ressourcen
 - überflüssige Entwicklungsressourcen
 - überflüssiges Administrationskompetenz benötigt

11

Pure Spring



- Pro
 - keine Ressourcenverschwendung
 - **Laufzeit und Speicher der Produktivumgebung**
 - **Administrationswissen**
 - **Entwicklerarbeitsplatz**
 - einfacher Integrations und Komponententest
 - **offeriert schnelle Entwicklungszyklen**
- Contra
 - kein shared Resources Environment
 - komplexe Konfiguration zur Bereitstellung spezieller API
 - **JTA, JMS, JCA**
 - kein zertifizierter Mix von Enterprise Service Komponenten
 - undefinierte QoS-Angebote
 - keine standardisierte Administrations- und Managementsicht der Anwendung

12

Warum nicht Tomcat als Application Server?



- „Spring im skalierbaren Webcontainer“
 - again - Spring beantwortet nicht alle Fragen im Lebenszyklus
 - JMS
 - JTA (XA)
- kein Erweiterungskonzept für Deployments
 - Spring Deployment
 - alternative Modularisierungsansätze
- Administrations und Managementunterstützung begrenzt
 - Datenbankpool verwalten
 - mangelndes KnowHow bei Administratoren
- mangelnde Möglichkeiten der Konfiguration zur Laufzeit
- Softwareverteilungsprobleme
- optionale EJB Unterstützung ?
 - nur in Anwendung (z. B. OpenEJB)

13

Alternative Geronimo?



- Pro
 - „Geronimo And Tomcat Provide The Core Platform“
 - „ASF Projects Create Open Source Software For SOA And Web Services“ 3/2007 Forrester
 - Websphere Community Edition
 - “Strong Performer in Open Source Projects“ Q2/2006 Forrester
- Contra
 - J(2)EE Risiko und Potential als Transaktionsplattform niedrig (Gartner 2/2007)
 - Apache Geronimo Nischenprodukt „Magic Quadrant for Enterprise Application Servers,2Q06“Gartner

14

Agenda

- Einführung
 - Motivation (J2EE bis heute)
 - Problem
 - **Geronimo**
- Szenarien
 - Konfiguration- und Installationsoptionen
 - Mandantenfähigkeit



15

Geronimo? Was ist das...

- J(2)EE konformer Application Server
 - J2EE 1.4 TCK Geronimo 1.0-M 5 Oktober `05
 - JEE 5.0 TCK Geronimo 2.0-M6 Juni `07
- OpenSource Implementierung von Apache
 - Steht unter Apache Software License nicht LGPL
- „Zusammenstellung“ mehrerer Projekte
 - Vervollständigung des J2EE Stacks durch ASF/BSD lizenzierten Code
- Erstellung eines „eigenen Servers“ möglich

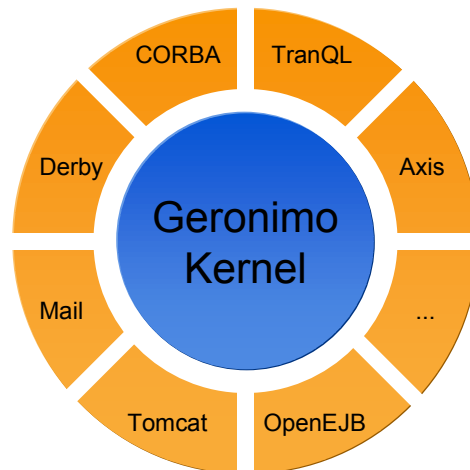
Warum Application Server von Apache?

- Bisher kein J2EE Server mit BSD abgeleiteter Lizenz
 - JBoss und JOnAS haben GPL abgeleitete Lizenz
- Apache Software Foundation hat einige J2EE Projekte
 - einzelne Bestandteile aber keine Integration
- Best of breed app server reusing Open Source from the community



17

Geronimos Integrationsgedanke



18

Agenda

- Einführung
 - Motivation (J2EE bis heute)
 - Problem
 - Geronimo
- **Szenarien**
 - Konfiguration- und Installationsoptionen
 - Mandantenfähigkeit



19

Anforderungen für den Live Betrieb - OIO Kundensicht I

- Konfiguration des Servers muss einfach sein!
 - Administratoren wollen nicht „kryptische“ XML Dateien editieren
- Aufsetzen eines weiteren Servers muss einfach sein
 - ebenso Update auf neue Serverversion auf mehreren Instanzen
 - „Cloning“ der Umgebung
 - Softwareverteilung, Updatemechanismus
 - „Cluster-Update“
- Stabilität, Performace
 - Ausfallsicherheit - Standby System
- Mandantenfähigkeit, Multi-Application Environment

20

Anforderungen für den Live Betrieb - OIO Kundensicht II



- Erweiterung zur Laufzeit
 - z. B. auch neue Bibliotheken in Classpath
- Complete Deployment
 - ein Archiv pro Anwendung
- Management - Überwachung
 - Einfache Tools, JMX, etc
- Sicherheit
 - Standardmäßig nicht „alles offen“

21

Optimierte Laufzeitumgebung...



- Keine ungenutzten Komponenten
 - EJB Container
 - JCA-Adapter
 - JMS-Provider
- Keine fragwürdigen Standardeinstellungen
 - Einstellungen meist anwendungsabhängig!
- Eventuell Erweiterbarkeit

22

Agenda

- Einführung
 - Motivation (J2EE bis heute)
 - Problem
 - Geronimo
- Szenarien
 - **Konfiguration- und Installationsoptionen**
 - Mandantenfähigkeit



23

Geronimo Server und Module

- Server arbeitet modulbasiert
 - Modul Verwaltungseinheit des Servers
 - **elementare, zusammengehörende Bausteine (z. B. Connector)**
- Serverbestandteile sowie Applikationen sind Module
 - können auch „zusammen gepackt“ werden
 - **z. B. EJB Anwendung mit konfigurierter Datenbankanbindung**
- Module müssen nicht ausgeführt werden
 - können sich „offline“ im Server befinden
 - **Stop bedeutet kein Undeployment**

24

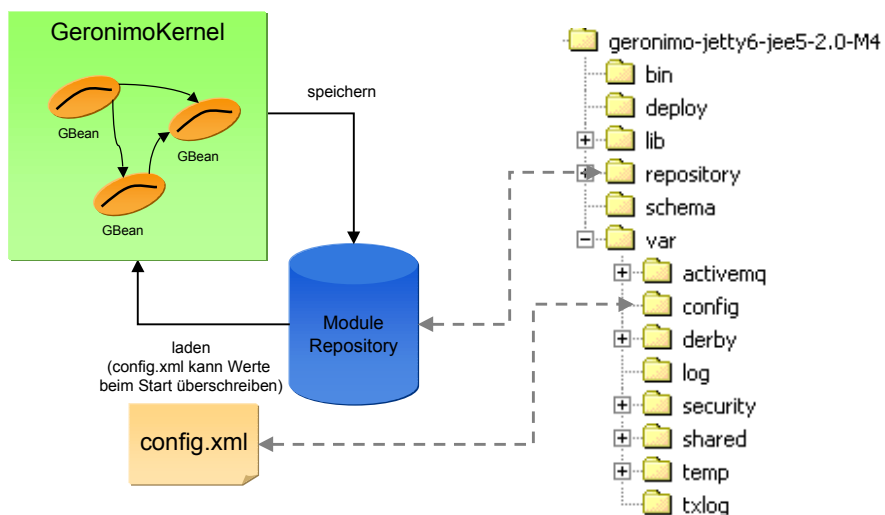
Geronimo Module - (II)

- Modul besitzt eindeutigen Namen innerhalb des Servers (ModuleId)
 - Beispiel: `geronimo/welcome-tomcat/1.1-SNAPSHOT/car`

}	}	}	}
groupId	artifactId	version	type
- Innerhalb des Servers Speicherung in Repository
 - Ablage erfolgt im proprietärem CAR Format
 - CAR Datei enthält Konfigurationsinformationen
- Module Zusammenstellung von konfigurierten GBeans
 - GBeans kleinste verwaltbare Einheit

25

Geronimo Modulverwaltung- und Konfiguration



26

Einfache Konfiguration mittels var/config/config.xml



- Zentrale Konfigurationsdatei für alle Module (config.xml)
 - Werte aus CAR im Repository können angepasst werden
 - **Portumstellungen, etc**
- Schnelle Anpassung an Umgebung möglich
 - Gleiches Modul mit Anpassungen in verschiedenen Servern
 - Platzhalterersetzungen in config.xml möglich
 - **config-substitutions.properties**
 - Ports umstellen mit Offset (portOffset)
- Änderungen zur Laufzeit werden persistiert
 - keine Änderung in Datei zur Laufzeit möglich!

27

Beispiel: Arbeiten mit Modulen



```
$/deploy.bat list-modules
```

```
$/deploy.bat stop geronimo/welcome-tomcat/1.1.1/car
```

```
<attributes xmlns="http://geronimo.apache.org/xml/ns/attributes-1.1">
  <module name="org.apache.geronimo.configs/rmi-naming/2.0-M4/car">
    <gbean name="RMIRegistry">
      <attribute name="port">1099</attribute>
    </gbean>
    <gbean name="NamingProperties">
      <attribute name="namingProviderUrl">rmi://0.0.0.0:1099</attribute>
    </gbean>
    ...
  </module>
  ...
```

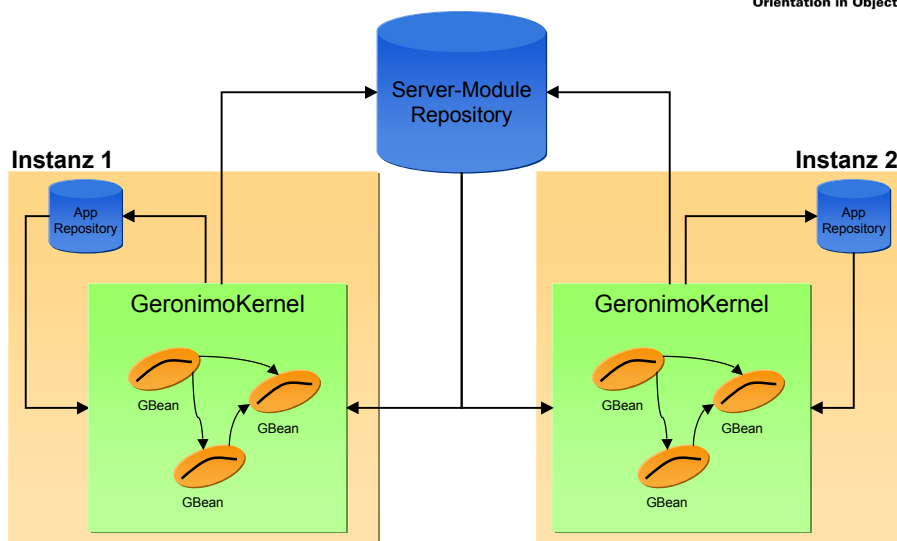


Multiple Serverinstances - Geronimo 2.0

- Unterstützung für mehrere Instanzen aus einer Installation
 - nur Konfiguration unterscheidet sich
 - Repository wird geteilt
 - **Multiple Repositories möglich!**
- Servername muss beim Start angegeben werden
 - Name bezieht sich auf Verzeichnis, in dem var Dir gesucht wird
 - **org.apache.geronimo.server.name=servers/appServer1**
 - „Gleiche Anwendungen - Unterschiedliche Konfiguration“
- Portkonflikte können einfach behoben werden
 - Beim Start portOffset mit angeben
 - config-substitutions.properties verwenden

29

Multiple Repositories - Geronimo 2.0 I



30

Multiple Repositories - Geronimo 2.0 II

- Server kann mehrere Repositories besitzen
 - Trennung Server und Anwendungen
- Gleiche „Serverinstallation“ für mehrere Instanzen
 - nur Anwendungen unterscheiden sich
 - **schneller und komfortabler Update über „alle Server“**
- Ziel-Repository kann beim Deployment angegeben werden
 - Standardmäßig Deployment in alle Repos
 - leider nur Kommandozeilenunterstützung
 - **--target Option bei Deployer skript**

31

Agenda

- Einführung
 - Motivation (J2EE bis heute)
 - Problem
 - Geronimo
- Szenarien
 - Konfiguration- und Installationsoptionen
 - **Mandantenfähigkeit**



32

Mandantenfähigkeit - 2 Anwendungen



- Möglichkeit 1
 - 2 Anwendungen auf 2 Serverinstanzen auf 2 Rechnern
 - **oft Realität - teuer und schwer wartbar**
- Möglichkeit 2
 - 2 Anwendungen in *jeweils* 1 Serverinstanz auf 1 Rechner
 - **siehe „Multiple Serverinstances“/ „Multiple Repositories“**
- Möglichkeit 3
 - 2 Anwendungen auf 1 Serverinstanz auf 1 Rechner
 - **So war und ist JavaEE Spec doch gemeint, oder? ;-)**
 - **oft der Wunsch aber nicht umgesetzt**

33

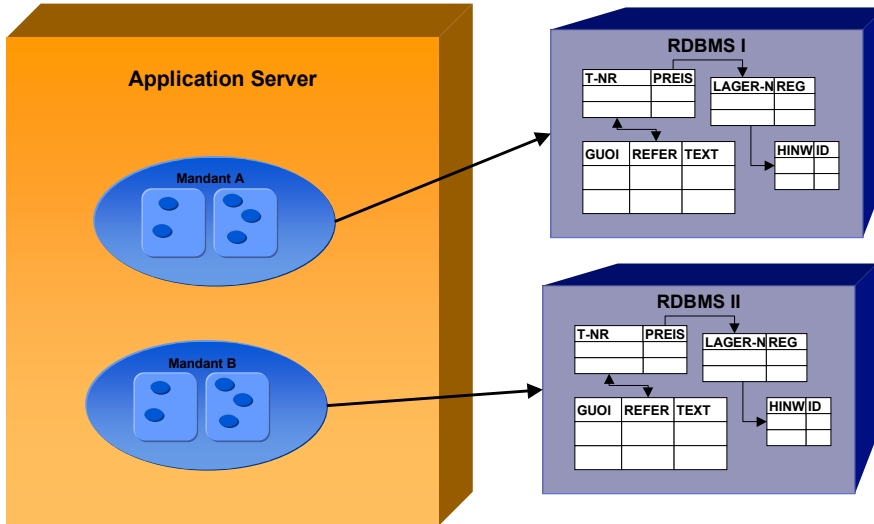
Der 3. Weg zur Mandantenfähigkeit



- Probleme wenn Classloading nicht „richtig“ umgesetzt
 - Anwendungen müssen getrennt voneinander laufen
 - **z. B. Singletons, unterschiedliche Lib Versionen**
- Oft Mandantenfähigkeit über Buildmanagement - 2 Archive
 - Anwendungen unterscheiden sich nur in Konfiguration
 - **z. B. andere DB, anderer Web-Context**
- Potenziell Probleme beim „Austausch“ einer Anwendung zur Laufzeit
 - Anwendung besitzt Abhängigkeiten zu Fremdbibliotheken
 - **Datenbankpool -> Treiber nicht im Classpath**

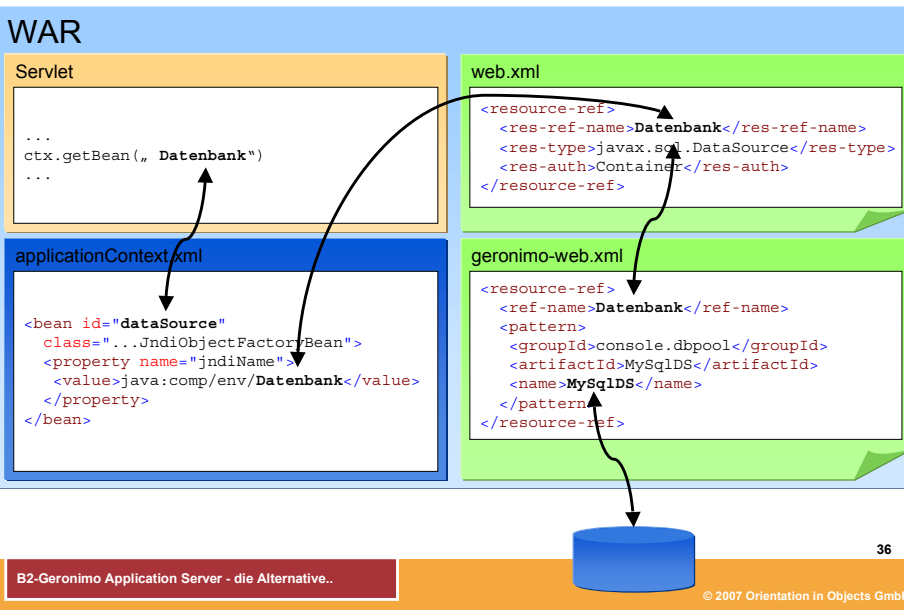
34

Beispiel: Mandantenfähige Anwendung



35

Beispiel: Anwendungsaufbau



36

Beispiel: Deployment - Möglichkeiten



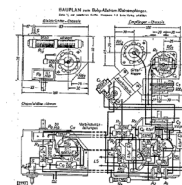
- Webanwendung greift auf Datenbank zu
 - Servlet holt über JNDI Lookup Referenz zur DB
 - Lookup „java:comp/env/Datenbank“
- Szenario 1 - „klassischer Fall“
 - Enterprise Archive (EAR) enthält Webanwendung
 - Datenbank separat innerhalb des Application Servers installiert
- Szenario 2
 - Enterprise Archive (EAR) enthält Webanwendung **und** ConnectionPool
 - Datenbank wird **mit** der Anwendung installiert und verwaltet

37

Konfigurationen erstellen - Deployment Plan



- XML Datei zur Modulbeschreibung
 - für Dienste, Ressourcen und Anwendungen
 - z. B. PortNummer des WebServers, Datenbankangabe für EJBs
 - besitzt eindeutigen Namen
- Deploymentprozess benötigt einen Deployment Plan
 - im Archiv oder als Parameter für Deployer
 - META-INF/geronimo-application.xml
- Muss nicht mit eingepackt werden!
 - Andere Server erfordern Bundle
- Kann „mehrere Archive“ auf einmal beschreiben
 - Komplette EAR Beschreibung



Beispiel: Anwendungsszenario 2



Kompletter Geronimo Deployment Plan für EAR

```
<?xml version="1.0"?>
<application ...>
  <sys:environment>
    <sys:moduleId>...</sys:moduleId>
    <sys:dependencies>
      <sys:dependency>...</sys:dependency>
    </sys:dependencies>
  </sys:environment>

  <module>
    <web>db-web-application-1.0.war</web>
    <web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.2">
      ... Inhalt der geronimo-web.xml ...
    </web-app>
  </module>
  <module>
    <connector>tranql-connector-ra-1.3.rar</connector>
    <connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector-1.2">
      ... Inhalt der geronimo-connector.xml ...
    </connector>
  </module>
</application>
```

40

Ausblick Geronimo Assemblies

- Verschiedene Server-assemblies direkt unter Apache Distribution
 - <http://geronimo.apache.org/downloads.html>
 - Einsatzprofile (siehe Problemstellung)
- interessante Projekte
 - GASwerk
 - <http://gaswerk.sourceforge.net/>



- Standardisierung in JEE 6.0 ?

41



Vielen Dank für Ihre Aufmerksamkeit !

Für weitere Fragen wir stehen Ihnen gerne auch im Tagesverlauf auf Stand 31 zur Verfügung.

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0