JAVA DAYS ´99



Entwicklung von Enterprise JavaBeans

Klaus Krüger
Andreas Liebscher
MGM EDV-Beratung GmbH

Sun microsystems

Inhaltsverzeichnis

1 Einführung

1.1 Was ist „Enterprise JavaBeans“?

1.2 EJB-Architektur

2 Entwicklung von Enterprise JavaBeans

2.1 Entwicklungszyklus

2.2 SessionBeans

2.3 EntityBeans

3 Zusammenfassung

© 1999 Sun Microsystems GmbH

Was ist „Enterprise JavaBeans“?



- Ein Komponentenmodell

> Enterprise JavaBeans ist ein
> *Server-Komponentenmodell*
> für eine
> *verteilte* und *mehrschichtige*
> *objekt-orientierte Architektur*

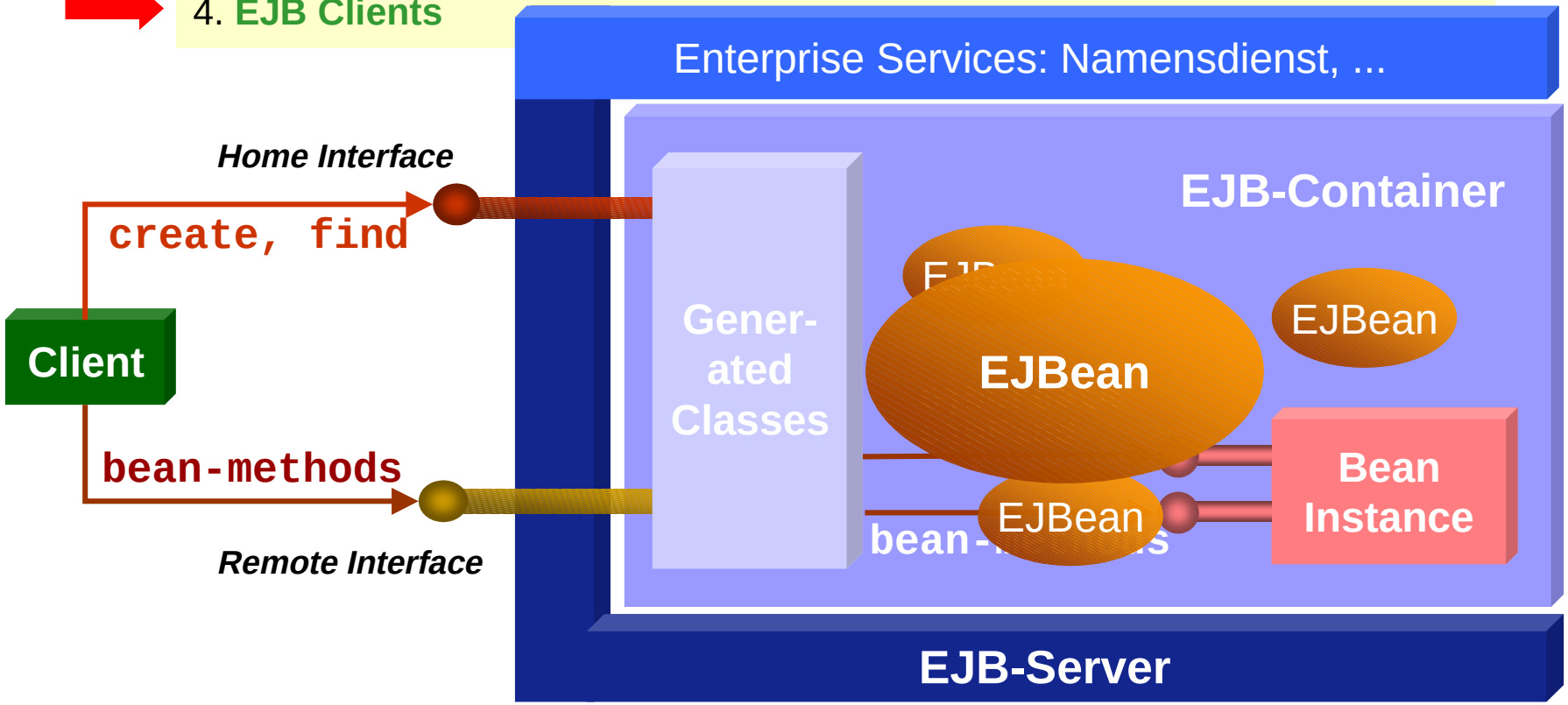
- Eine Spezifikation
  - Sie legt die Architektur und das API fest
  - Wird von Sun unter Beteiligung vieler Hersteller erarbeitet (u.a. IBM, Oracle, Siemens, Sybase, Inprise, BEA, ...)
  - Ist ein Industriestandard, also kein Produkt

© 1999 Sun Microsystems GmbH

Sun microsystems

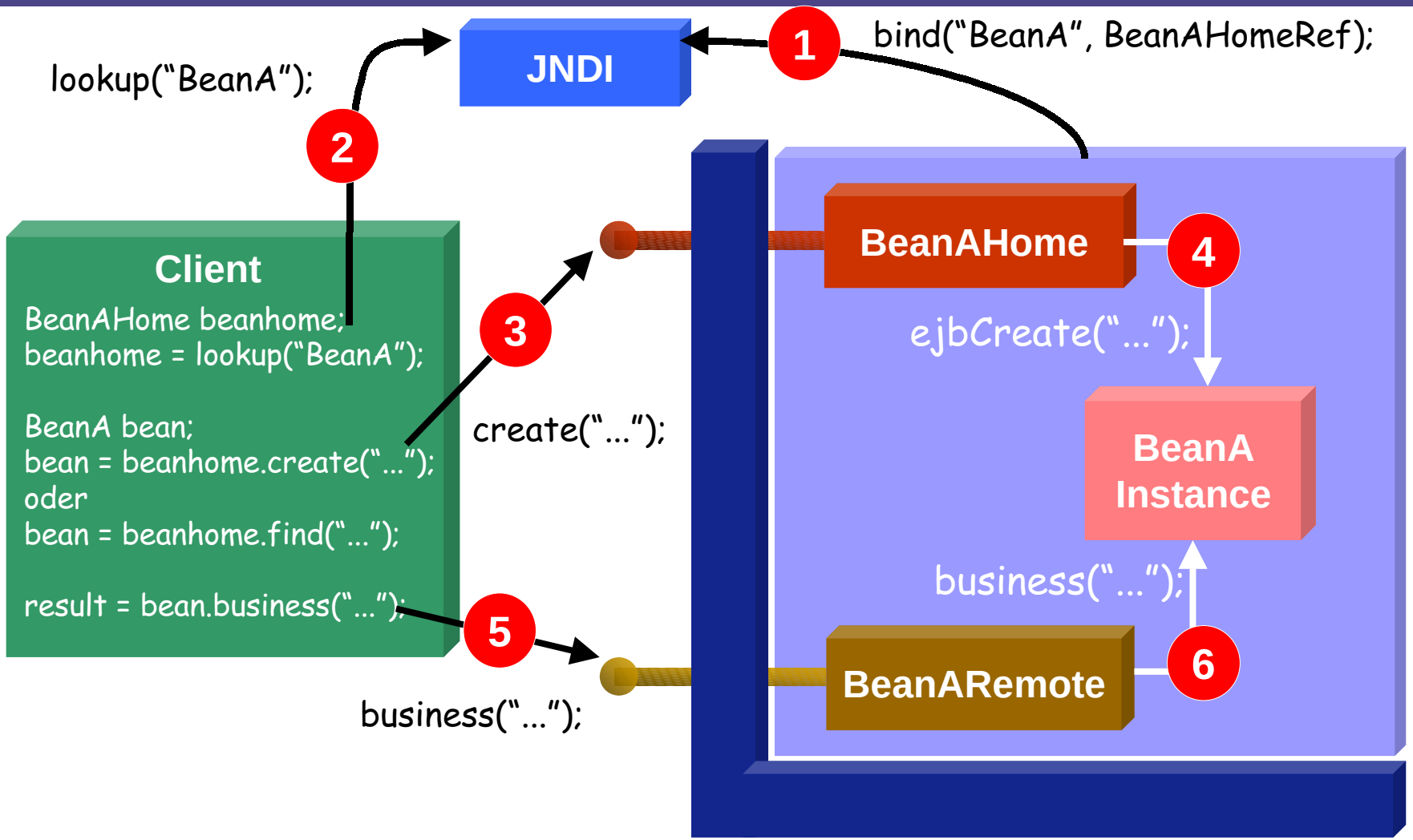
Architekturelemente von EJB

1. Infrastruktur: **EJB-Server**, **EJB-Container**, **Enterprise Services**
2. EJB Objekte: **Generated Classes**, **Bean Instance**
3. EJB Interfaces: **Home Interface**, **Remote Interface**
4. **EJB Clients**



© 1999 Sun Microsystems GmbH

EJB – Szenario



© 1999 Sun Microsystems GmbH

Bean-Typen



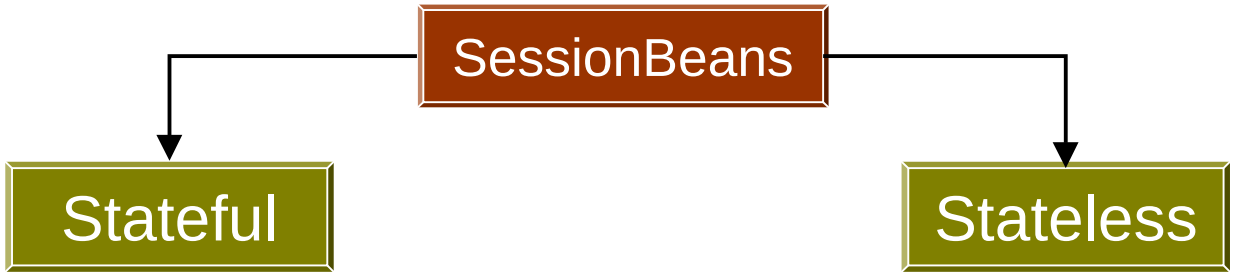
SessionBeans		EntityBeans
Modellieren Prozesse & Abläufe	⟷	Repräsentieren Geschäftsdaten
- Zugriff auf Datenbank		- verknüpft mit Datenbank
- z.B. Abfrage des Kontostands		- z.B. Daten eines Bankkontos
Einem Client zugeordnet	⟷	Von mehreren Clients nutzbar
- anonym		- besitzen Identität
Transiente Objekte	⟷	Persistente Objekte
- kurzlebig, für eine Sitzung		- langlebig
- überleben Systemabsturz nicht		- überleben Systemabstürze

© 1999 Sun Microsystems GmbH

Sun microsystems

SessionBeans: Stateful vs. Stateless

- Transient
- Mit oder ohne „Konversationsgedächtnis“
  (Client-spezifischer Zustand zwischen Methodenaufrufen):



mit Konversationsgedächtnis:	⟷	ohne Konversationsgedächtnis
Bean-Instanz ist genau **einem Client** zugeordnet	⟷	Bean-Instanz kann **jedem Client** für einen Aufruf zugeordnet werden
kann „passiviert“ und wieder „aktiviert“ werden	⟷	Bean-Instanz kann problemlos freigegeben werden

© 1999 Sun Microsystems GmbH

EntityBeans: Container- vs. Bean-managed

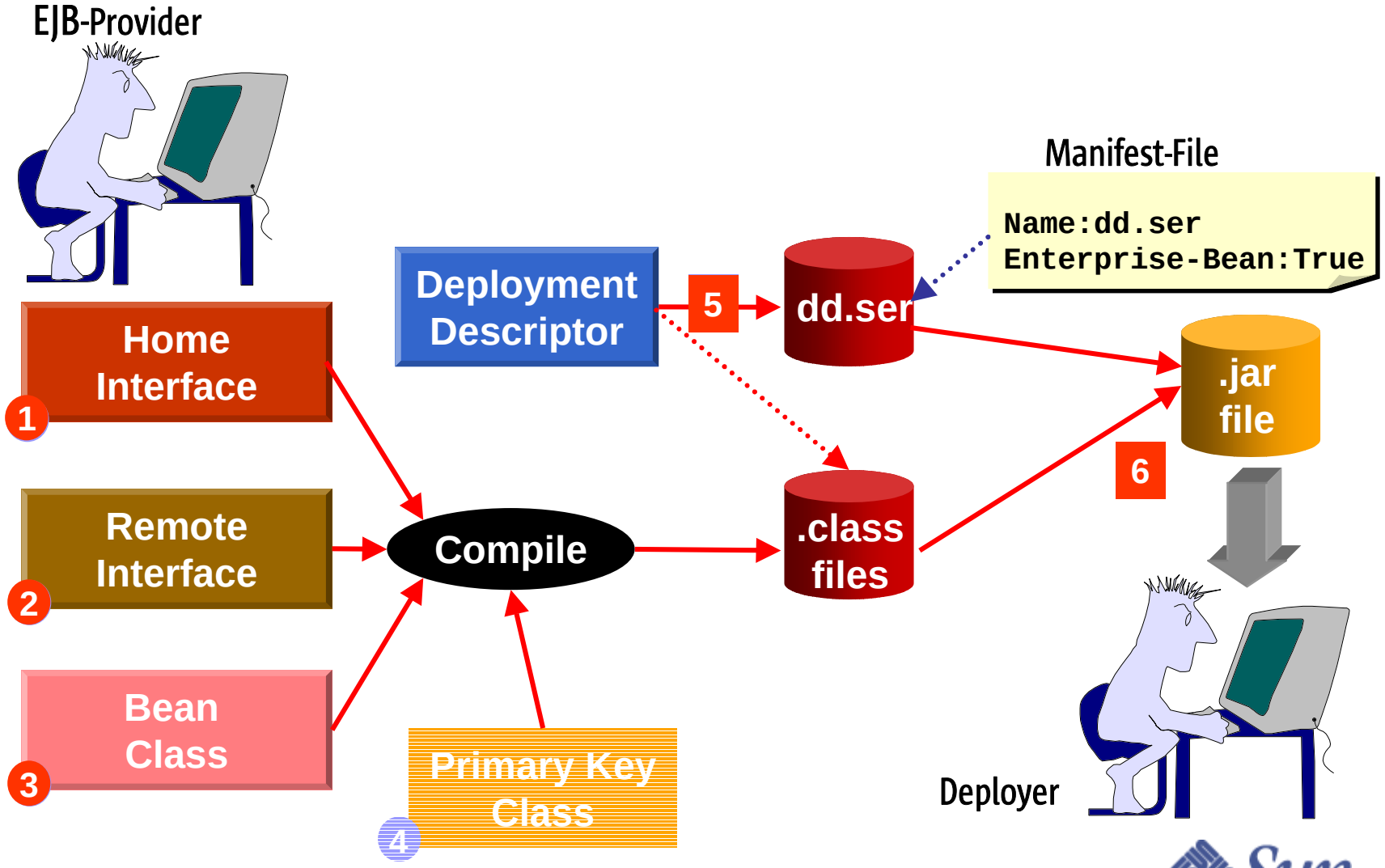
- Verwaltung des persistenten Zustands durch Bean oder Container:



Zustandssicherung durch Bean	⟷	Zustandsicherung durch Container
Bean-Provider codiert DB-Aufrufe (als Teil der Bean-Business-Logic)	⟷	Container generiert DB-Aufrufe (beim Deployment)
		Deployment Deskriptor spezifiziert zu sichernden Bean-Attribute

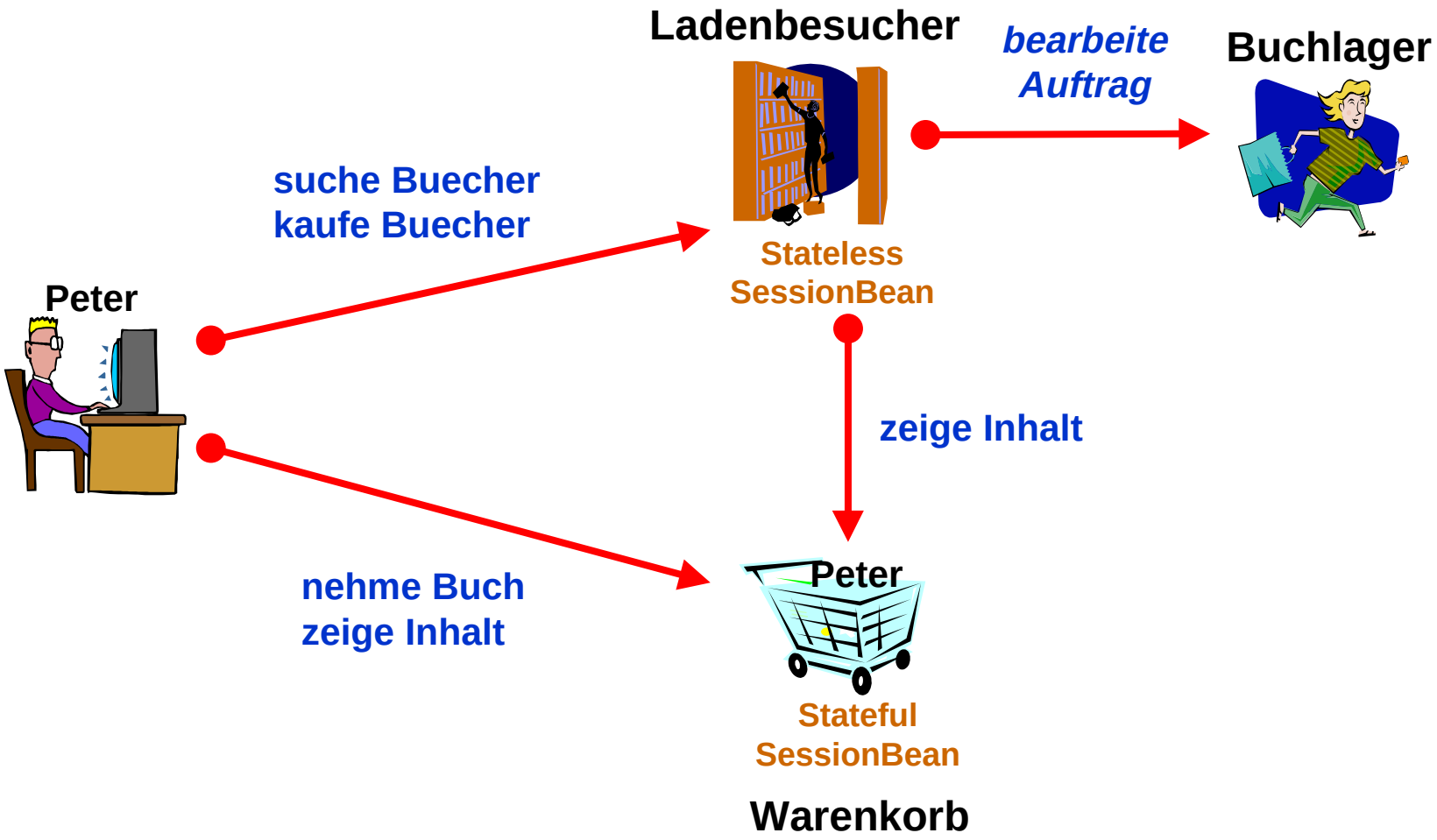
© 1999 Sun Microsystems GmbH

Entwicklungsschritte SessionBean



© 1999 Sun Microsystems GmbH

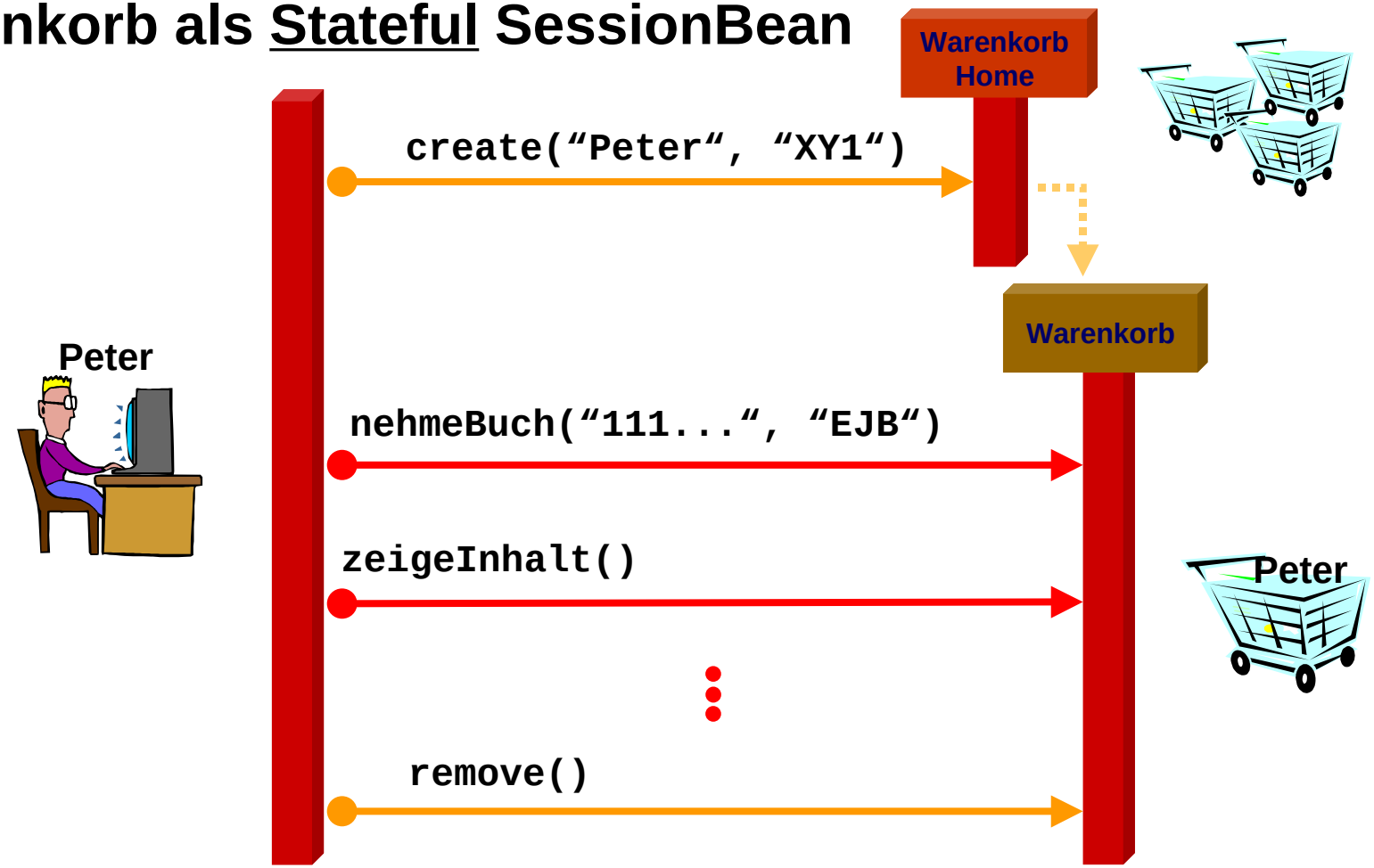
Online-Shopping - Szenario



© 1999 Sun Microsystems GmbH

Online-Shopping

Warenkorb als Stateful SessionBean



© 1999 Sun Microsystems GmbH

1 Home Interface ome-Schnittstelle



- Erweitert die EJBHome-Schnittstelle
- Deklariert eine oder mehrere **`create`**-Methoden
  - Erzeugt eine neue Bean-Instanz
  - Argumente dienen zur Initialisierung des Beans
  - Stateless SessionBeans dürfen keine **`create`**-Methode mit Argumenten haben

```
package OnlineShopping;

import javax.ejb.*;
import java.rmi.*;

public interface WarenkorbHome extends EJBHome {
    public Warenkorb create(String login, String passwort)
                        throws RemoteException, CreateException;
}
```


© 1999 Sun Microsystems GmbH

2 Remote Interface emote-Schnittstelle



- Erweitert die EJBObject-Schnittstelle
- Deklariert die Business-Methoden
  - Entsprechend den RMI-Regeln
  - Methoden müssen **`RemoteException`** deklarieren
  - Jede Methode hat eine entsprechende Methode in der Bean-Klasse

```
package OnlineShopping;

import javax.ejb.*;
import java.rmi.*;

public interface Warenkorb extends EJBObject {
   public void      nehmeBuch(String isbn, String titel)
                                 throws RemoteException;
   public BuchListe zeigeInhalt()
                                 throws RemoteException;
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

3 Bean Class ean-Klasse

- Implementiert die SessionBean-Schnittstelle mit folgenden Methoden
  - setSessionContext() – nach dem Erzeugen der Bean-Instanz
  - ejbRemove() – vor dem Enfernen
  - ejbActivate() – nach Aktivierung
  - ejbPassivate() – vor Passivierung
- Zu jeder Business-Methode der Remote-Schnittstelle existiert eine entsprechende Methode
- Zu jeder **`create`**-Methode der Home-Schnittstelle muß eine entsprechende **`ejbCreate`**-Methode existieren

```
public class WarenkorbImpl implements SessionBean {

   << Verwaltungsmethoden (Kontrakt Bean-Container)>>
   << Business-Methoden >>
   << Konstruktoren >>
}
```


© 1999 Sun Microsystems GmbH

Code der Bean-Klasse

```
package OnlineShopping;
import javax.ejb.*;
import java.rmi.*;

public class WarenkorbImpl implements SessionBean {
     protected SessionContext  fContext;
     protected String          fLogin;
     protected BuchListe       fBuchListe;

     public void setSessionContext(SessionContext context) throws RemoteException
          {fContext = context;}
     public void ejbActivate   () throws RemoteException {}
     public void ejbPassivate  () throws RemoteException {}
     public void ejbRemove     () throws RemoteException, RemoveException {}


     public void ejbCreate(String login, String passwort) throws RemoteException, CreateException {
          if (!checkLogin(login, passwort))
               throw CreateException("Sie sind nicht als Mitglied registriert!");
     }

     public void nehmeBuch(String isbn, String titel) throws RemoteException
          {fBuchListe.insert(isbn, titel);}
     public BuchListe zeigeInhalt() throws RemoteException
          {return fBuchListe.clone();}
}
```

"Konversations gedächtnis"

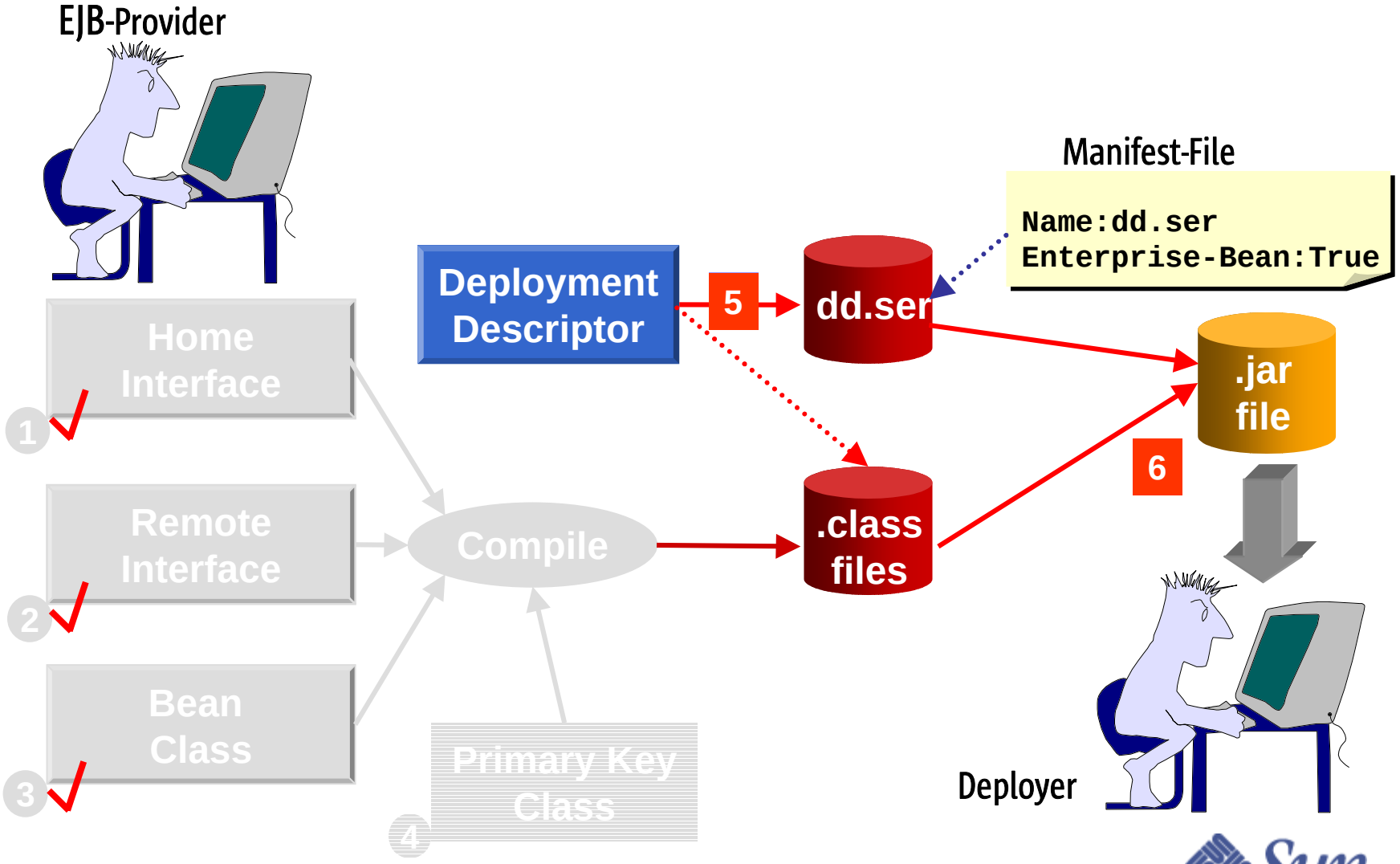
Lifecycle-Methoden

Konstruktoren

Business-Methoden

© 1999 Sun Microsystems GmbH

Entwicklungsschritte



© 1999 Sun Microsystems GmbH

5 Deployment Descriptor eployment-Deskriptor



Deployment-Deskriptor von Besucher

Name der Home-Schnittstelle	BesucherHome
Name der Remote-Schnittstelle	Besucher
Name der Bean-Klasse	BesucherImpl
Name des Home-Objektes	Shopping/Besucher
Transaktionsattribute	TX_BEAN_MANAGED
Security Attributes	Role: ALL Method: *
Art der Zustandsverwaltung	STATELESS_BEAN
Umgebungsvariablen	Name = Visitor ...

© 1999 Sun Microsystems GmbH

Sun microsystems

6 .jar-File ar-Datei

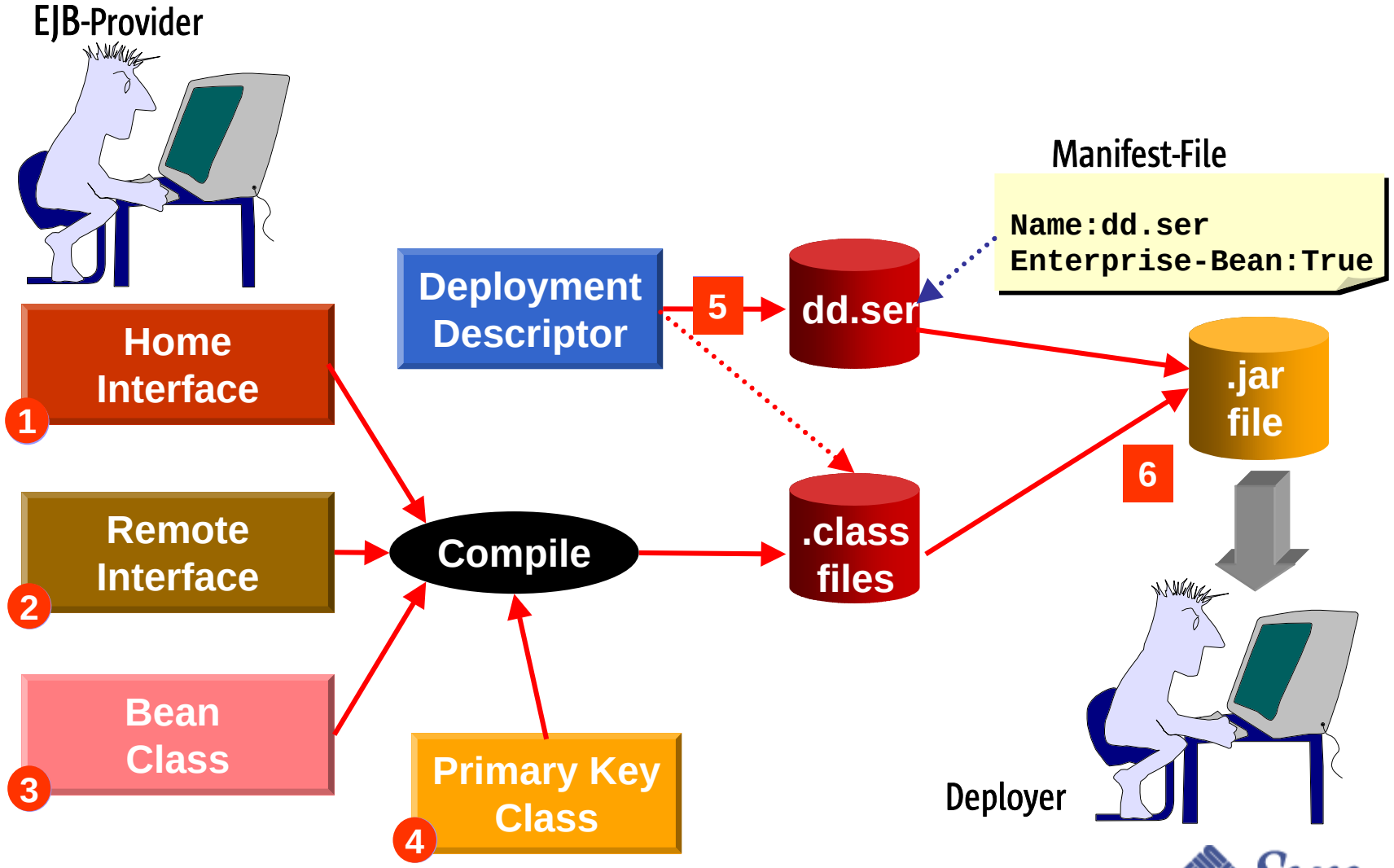


Serialisierter Deployment-Deskriptor

EJB 1.1: META-INF/ejb-jar.xml

© 1999 Sun Microsystems GmbH

Entwicklungsschritte EntityBean



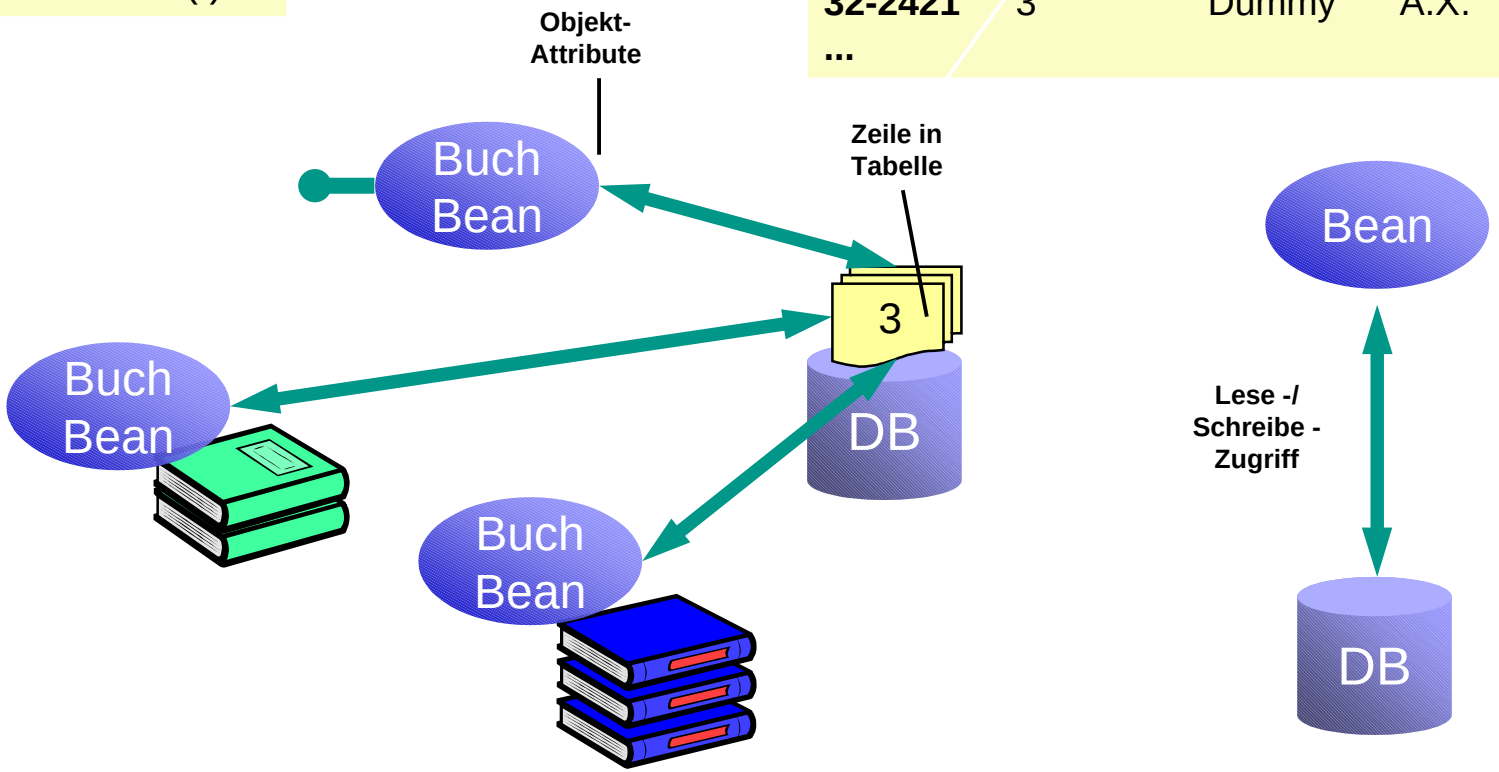
© 1999 Sun Microsystems GmbH

Online-Shopping - Buchlager

Buch-Lagerartikel als EntityBean

```
buchDaten()
imLagerVorhanden()
reduziereAnzahl()
```

ISBN	Anzahl	Titel	Autor
12-3323	2	EJB-Intro	M.B.
32-2421	3	Dummy	A.X.
...			



© 1999 Sun Microsystems GmbH

1 Home Interface ome-Schnittstelle



- Erweitert die EJBHome-Schnittstelle
- Enthält zum Suchen und Erzeugen von EntityBeans:
  - eine oder mehrere `create`-Methoden
  - die `findByPrimaryKey`-Methode zum Auffinden eines EntityBeans anhand des Primärschlüssels
  - evtl. weitere Suchmethoden, die mit dem Präfix `find` beginnen
  - eine `remove`-Methode

`find`-Methoden liefern eine oder mehrere EntityBeans-Objektreferenzen zurück

© 1999 Sun Microsystems GmbH

1 Home Interface ome-Schnittstelle




```
package OnlineShopping;

import javax.ejb.*;
import java.rmi.*;

public interface BuchHome extends EJBHome {

   Konto create(String isbn, String titel, String autor)
          throws RemoteException, CreateException;

   Konto create(String isbn, int anzahl, String titel, String autor)
          throws RemoteException, CreateException;


   Konto findByPrimaryKey (BuchPK isbn)
          throws RemoteException, FinderException;


   Enumeration findByAutor(String autor)
          throws RemoteException, FinderException;
}
```

Legen neue Bean-Instanz an (und dadurch Datensatz in DB)

Eindeutige Suche durch den Primärschlüssel

Liefert eine Menge von Entity-Bean-Referenzen zurück

© 1999 Sun Microsystems GmbH

2 Remote Interface emote-Schnittstelle



- Erweitert die EJBObject-Schnittstelle
- Definiert die Business-Methoden wie bei SessionBeans und stellt Sie dem Client zur Verfügung

```
package OnlineShopping;

import javax.ejb.*;
import java.rmi.*;

public interface Buch extends EJBObject {

   public BuchDaten  buchDaten()           throws RemoteException;
   public bool       imLagerVorhanden()    throws RemoteException;
   public void       reduziereAnzahl()     throws RemoteException;
}
```


© 1999 Sun Microsystems GmbH

3 Bean Class Bean-Klasse



- Implementiert die EntityBean-Schnittstelle mit folgenden Methoden
  - ejbLoad() – persistente Bean-Attribute aus DB lesen
  - ejbStore() – persistente Bean-Attribute in DB schreiben
  - setEntityContext() – nach dem Erzeugen der Bean-Instanz
  - unsetEntityContext() – vor dem Enfernen
  - ejbRemove(), ejbActivate(), ejbPassivate()
- Zu jeder **`create`**- und **`find`**-Methode der Home-Schnittstelle muß eine entsprechende **`ejbCreate`**- bzw. **`ejbFind`**-Methode existieren
- Zu jeder Business-Methode der Remote-Schnittstelle existiert eine entsprechende Methode

© 1999 Sun Microsystems GmbH

Sun microsystems

Code der Buch-Klasse




```
package OnlineShopping;
import javax.ejb.*;
import java.rmi.*;

public class BuchImpl implements EntityBean {

    public   BuchPK      attrISBN;
    public   int         attrAnzahl;
    public   String      attrTitel, attrAutor;

    protected    EntityContext    fContext;

    ...
    public bool  imLagerVorhanden()  throws RemoteException  {
        return (attrAnzahl > 0);
    }
    public void  reduziereAnzahl() throws RemoteException {
        attrAnzahl--;
    }
    ...
}
```

persistente Attribute

transiente Variablen

Business-Methoden

© 1999 Sun Microsystems GmbH

Sun microsystems

4 Primary Key Class rimärschlüssel



- Dient der eindeutigen Identifikation eines EntityBeans:
  - Zwei Beans mit gleichem Schlüssel und Home-Objekt gelten als identisch. Vergleich durch **`isIdentical()`**
- Wird durch eine serialisierbare Klasse mit folgenden Eigenschaften repräsentiert:
  - Alle Felder der Klasse sind public
  - Die Klasse selbst und der obligatorische Default-Konstruktor sind public
  - Die Methoden **`hashCode()`** und **`equals(Object other)`** sind in der Klasse implementiert

```
public class BuchPK implements java.io.Serializable {
    public String buchISBN;

    public BuchPK() {}
    public BuchPK(String isbn) {
        this.buchISBN = isbn;
    }
}
```


© 1999 Sun Microsystems GmbH

Find-Methoden



- Suchen nach Entitäten in der Datenbank
- Für jede **`findXXX`**-Methode der Home-Schnittstelle muß eine Methode **`ejbFindXXX`** implementiert werden

```
public Enumeration ejbFindAlleBuecher (String autor)
           throws FinderException, RemoteException {

    Enumeration buchEnum;
    try {
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery("Select ... from ... ");
        stmt.close();
        ...

    } catch (SQLException ex) {
        throw new FinderException();
    }

    return buchEnum;
}
```

Datenbankabfrage

© 1999 Sun Microsystems GmbH

Sun microsystems

5 Deployment Descriptor Deployment-Deskriptor



Enthält für ein EntityBean neben den allgemeinen Bean-Attributen noch folgende spezielle Attribute:

- Name der Klasse des Primärschlüssels (`PrimaryKeyClassName`)
- Liste der persistenten Felder (`ContainerManagedFields`)

© 1999 Sun Microsystems GmbH

Zusammenfassung



- EJB ist ein Serverkomponentenmodell
- Deklarative Spezifikation der Transaktionssteuerung, Security, Zustandsverwaltung und Persistenz
- SessionBeans
  - stateless
  - stateful
- EntityBeans
  - bean-managed
  - container-managed persistence
- Für die Unterstützung des Modells benötigt man
  - Applikationsserver mit EJB-Container
  - Container-Tools

© 1999 Sun Microsystems GmbH

Sun microsystems

EJB Quellen



- EJB SIG: Special Interest Group “Enterprise JavaBeans” der Java User Group Deutschland
  - Homepage: **`http://www.mgm-edv.de/ejbsig`**
  - Inhalt
    - Aktuelle Nachrichten zu EJB
    - Bestellmöglichkeit für den EJB SIG Newsletter
    - Produktübersichten zu
      - EJB-Server Implementierungen
      - EJB-Komponenten
      - sonstiger Software zu EJB
    - EJB-Workshops
    - Links zu vielen weiteren EJB-Quellen

© 1999 Sun Microsystems GmbH

Sun microsystems

Kontakt



Email: **Klaus.Krueger@mgm-edv.de**
oder **Andreas.Liebscher@mgm-edv.de**

MGM EDV-Beratung GmbH
Frankfurter Ring 105a
80807 München

Tel. +49 89 358680 - 0
Fax: +49 89 358680 - 88

© 1999 Sun Microsystems GmbH

Sun microsystems