

Web-Programmierung (WPR)

Vorlesung VII.

JavaServlets u. Java Server Pages (JSP)
Teil 2

<mailto:wpr@gruner.org>

14.9 JavaBeans und JSPs

- JavaBeans = Komponentenmodell von Java
- JavaBean ist eine Java-Klasse, wenn
 - getter/setter Methoden für Attribut-Zugriffe zur Verfügung stellt.
- JavaBean => Datencontainer für Darstellung

14.9.1 JSP-Aktionen

- Deklarative Aktionen
 - XML notiert
 - Standardisierte Tags
 - Einbindung von externen Tag-Libraries möglich
 - Eigene Tags möglich

14.9.1 JSP-Aktionen

- Zugriff auf JavaBeans von JSP
 - durch deklarative Aktionen (JSP-Tags)
 - Objekt erzeugen
 - `<jsp:useBean id="bezeichner" value="classname"/>`
 - Property lesen
 - `<jsp:getProperty name="strasse" value="classname"/>`
 - Property setzen
 - `<jsp:setProperty name="bezeichner"`
`property="propertyName" value="wert"/>`

14.9.2 JavaBeans und Benutzerdaten

Web-Browser

login.jsp

Account:

Password:

```
<form action="login.jsp">  
  <input name="name"/>  
  <input name="password"/>  
</form>
```

Web-Server / Servlet Engine

```
<jsp:useBean id="user" class="wpr.User" />  
<jsp:setProperty name="user" property="*" />
```

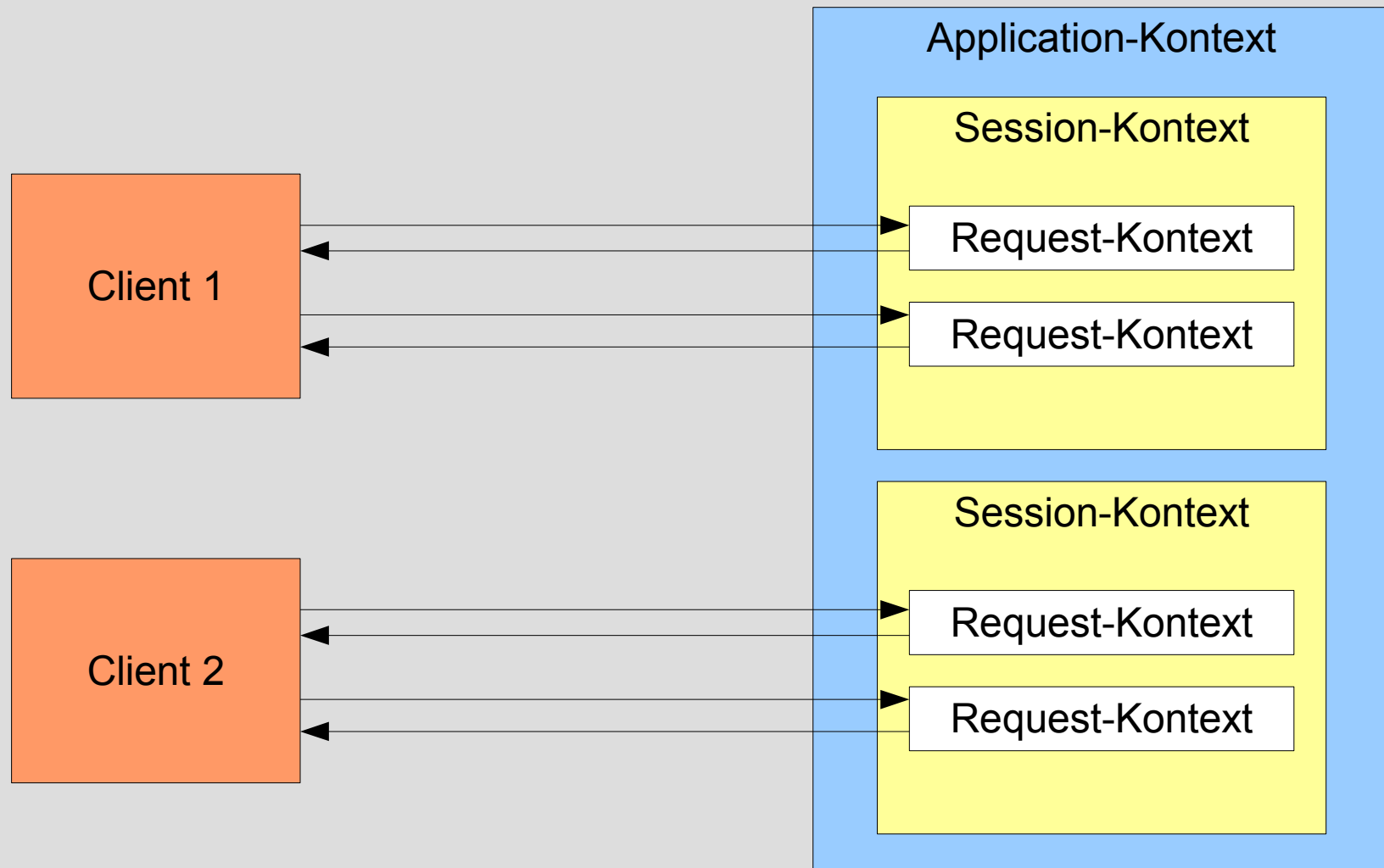
loginAction.jsp

User
-name : String
-password : String

14.9.3 Kontext (Scope)

- Daten werden kontextabhängig gespeichert
- Kontexte (von ServletEngine generiert)
 - ♦ Request-Kontext (request)
 - ♦ Session-Kontext (session)
 - ♦ Application-Kontext (application)
- „Kontext“-Zugriff
 - ♦ `setAttribute (String name, Object value)`
 - ♦ `getAttribute (String name)`

14.9.3 Kontext (Scope)



14.9.3 Kontext (Scope)

Wert	Bedeutung
page	das aktuelle Dokument
request	bean-Referenz wird im aktuellen Request-Kontext abgelegt
session	bean-Referenz wird im aktuellen Session-Kontext abgelegt
application	bean-Referenz wird im Application-Kontext (ServletContext) abgelegt und zwischen allen Servlets einer Applikation geteilt!

14.10 Verknüpfung von Servlets und JSPs

- Verarbeitung eines HTTP-Request durch Verknüpfung von Web-Komponenten
 - I. Weiterleiten an Web-Komponente
 - II. Einbinden der Ausgabe anderer Komponente
- **Servlet API – Klasse `RequestDispatcher`**
 - `forward (HttpServletRequest, HttpServletResponse)`
 - `include (HttpServletRequest, HttpServletResponse)`

14.10 Verknüpfung von Servlets und JSPs

- Servlet API – Klasse **RequestDispatcher**
 - Frage: Wo erhalte ich eine Referenz?

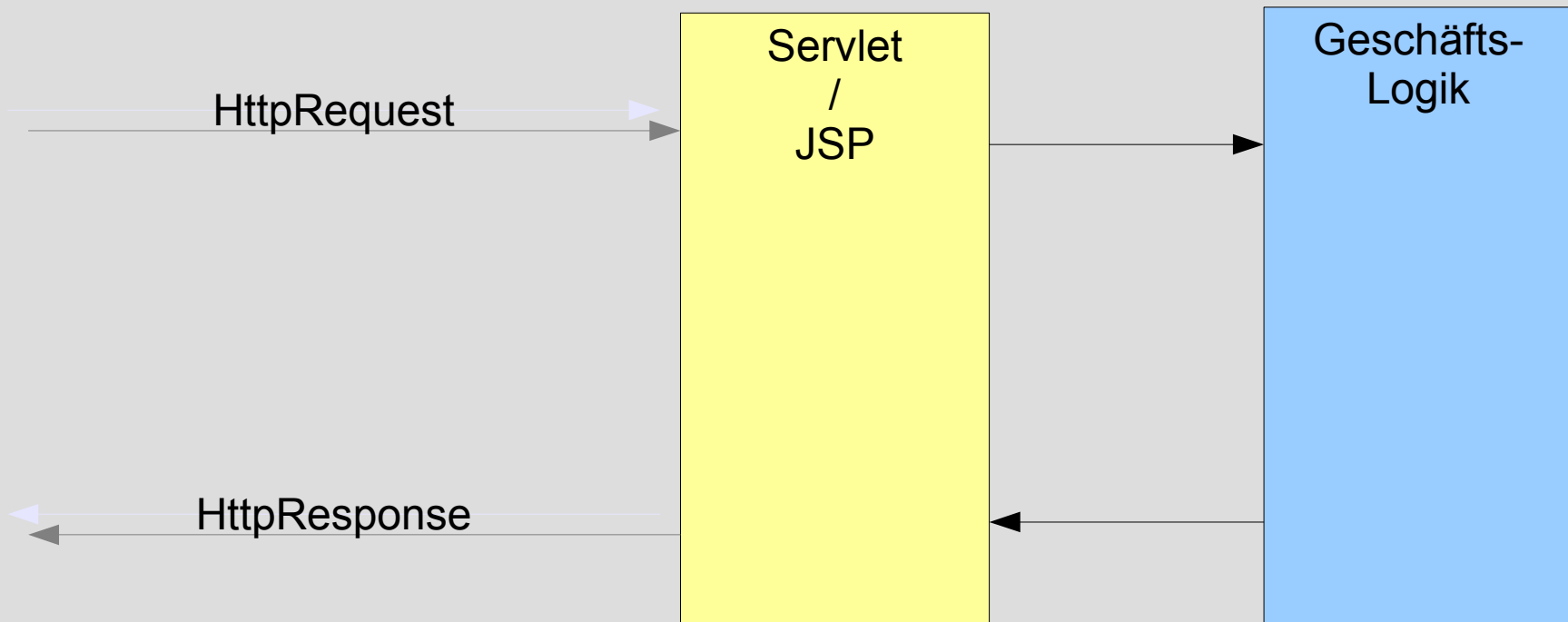
- Antwort 1: Servlet

```
RequestDispatcher dispatcher =  
    this.getServletContext().  
        getRequestDispatcher(targetURL);  
dispatcher.forward(request, response);
```

- Antwort 2: JSP

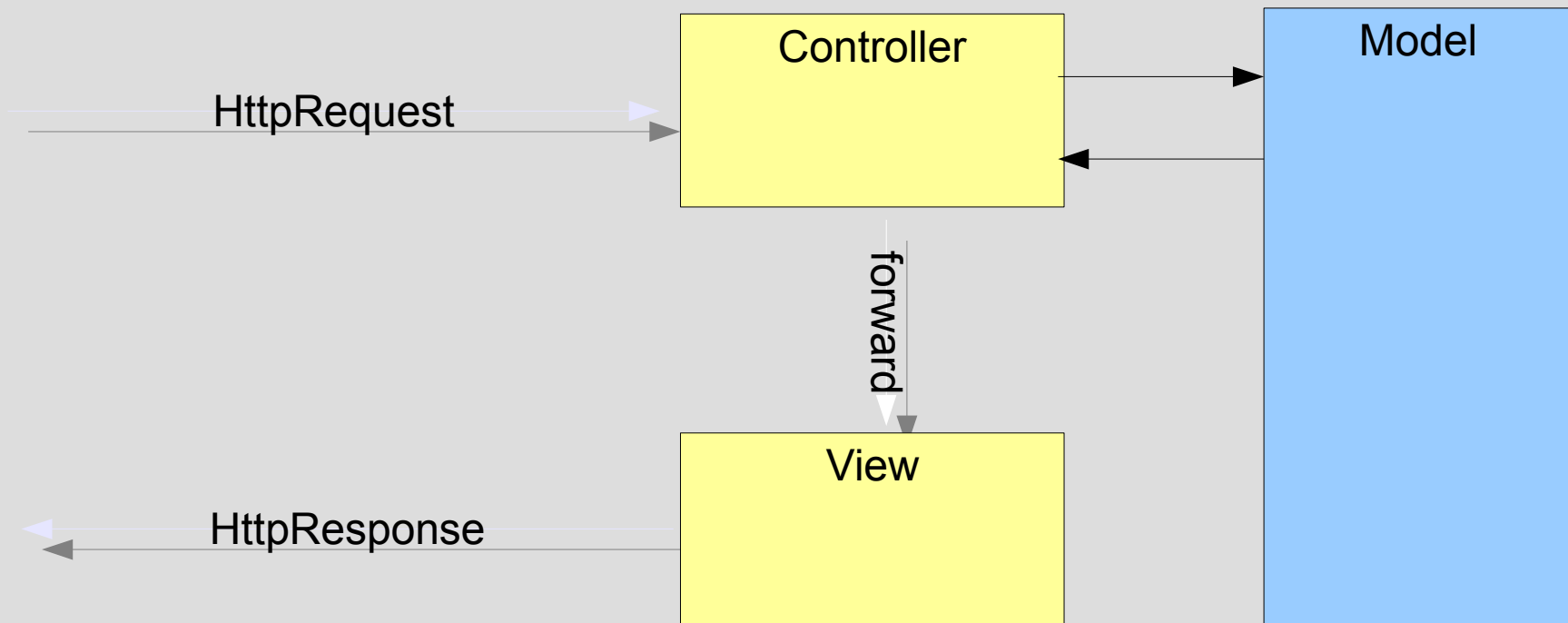
```
<jsp:forward page="targetURL"/>  
<jsp:include page="includeURL"/>
```

14.11 Programmiermodelle



14.11 Programmiermodelle

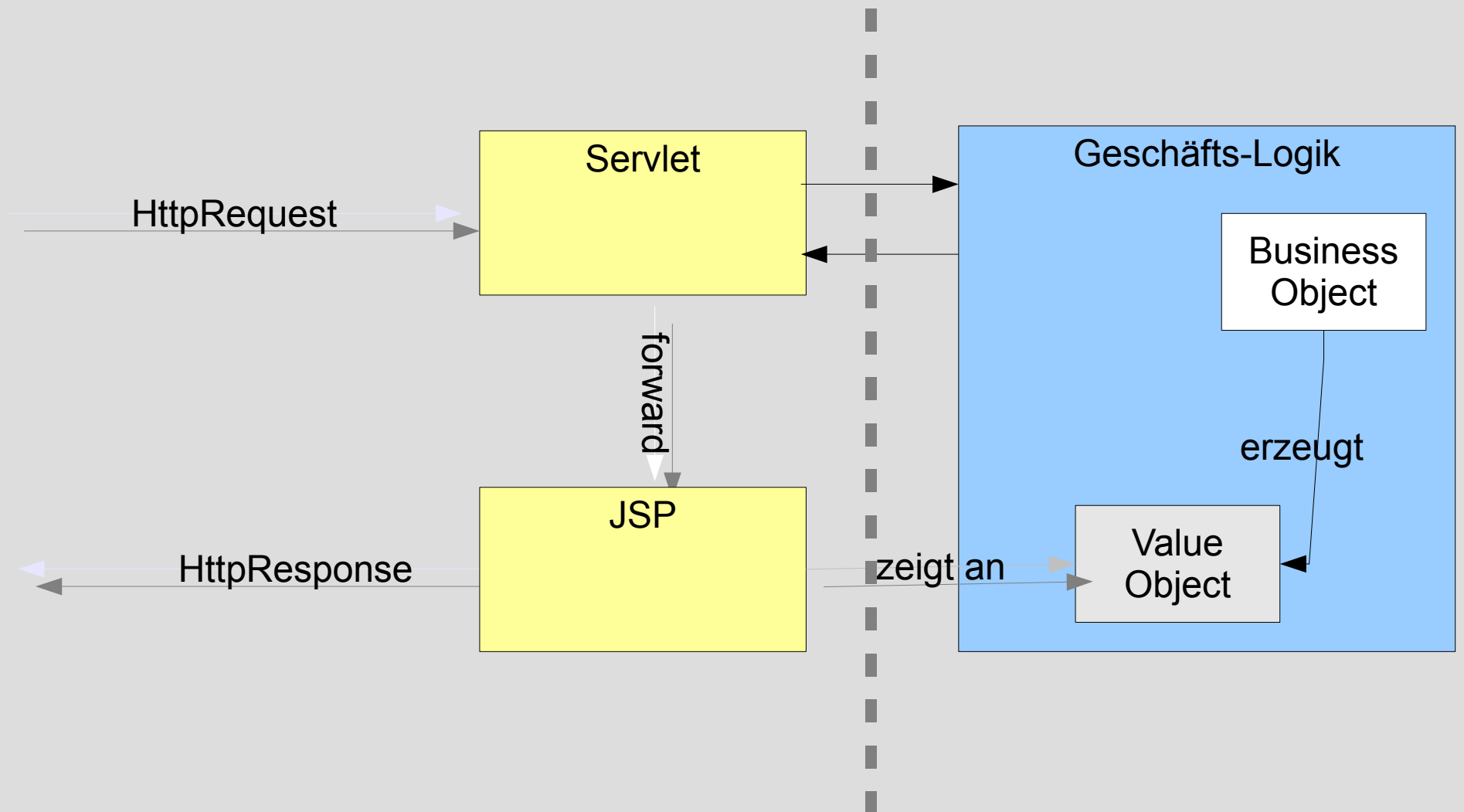
- Design Pattern: Model View Controller (MVC)



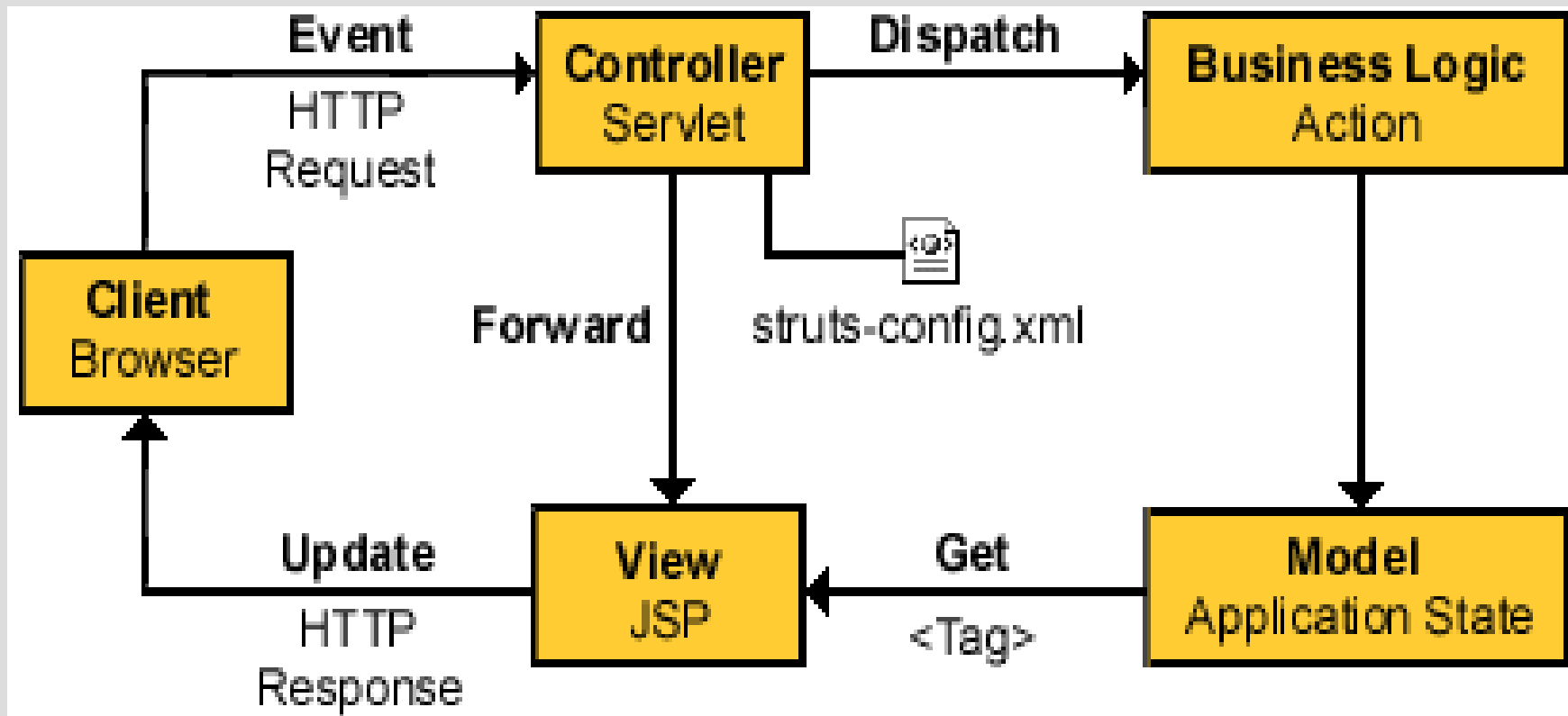
14.11 Programmiermodelle

- **Model**
 - ◊ Geschäftsdaten und deren fachliche Funktionalitäten
- **View**
 - ◊ Darstellung des Model
- **Controller**
 - ◊ Reagiert auf Benutzereingaben und verändert das Model

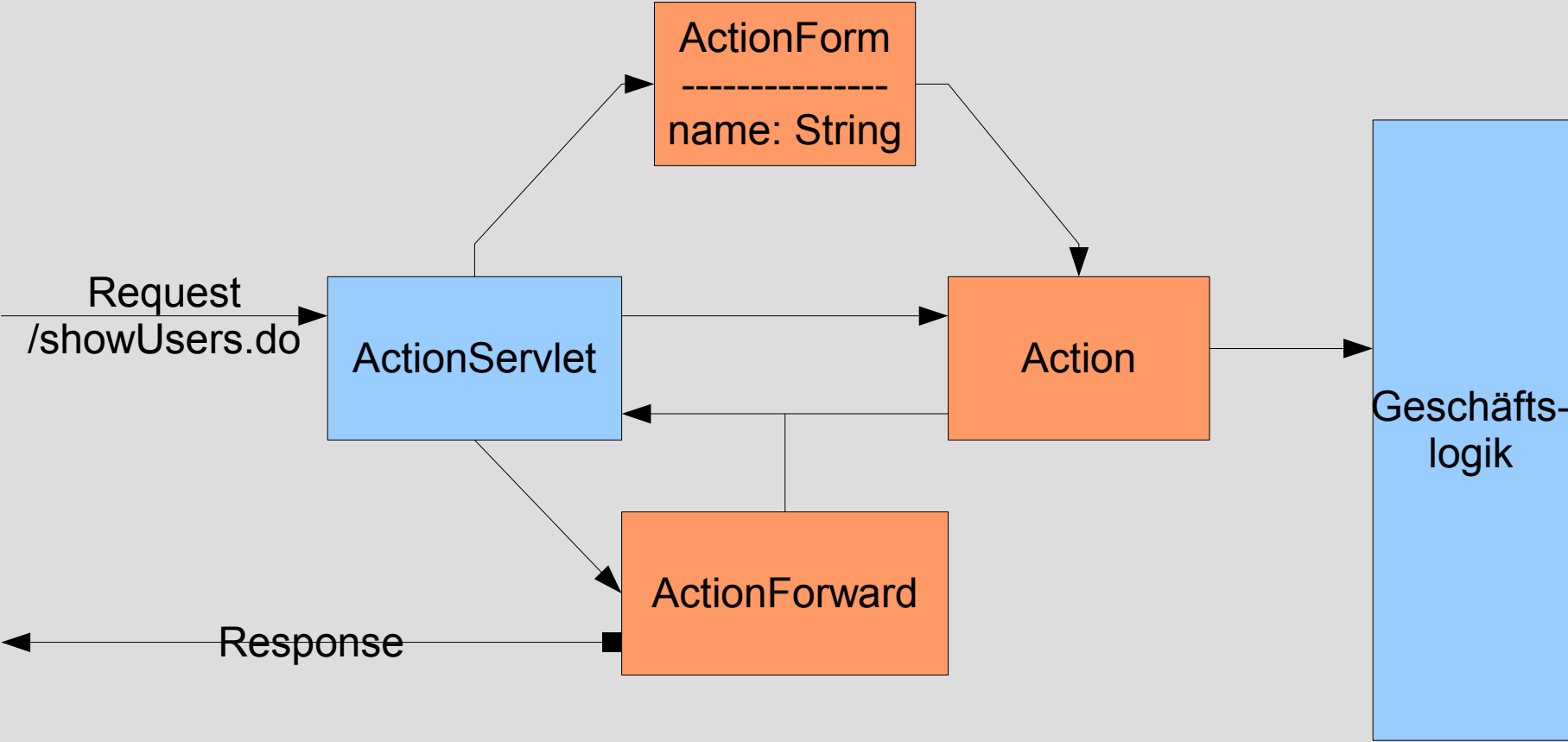
14.11 Programmiermodelle



14.X MVC mit Struts-Framework



14.X MVC mit Struts-Framework



14.X MVC mit Struts-Framework

- Struts-Konfiguration (struts-config.xml)
Action-Definition



```
<action path="/join"  
        name="joinForm"  
        type="web.mailinglist.JoinAction"  
        scope="request"  
        input="/joinMVC.jsp"  
        validate="true">  
    <forward name="success"  
            path="/welcome.html"/>  
</action>
```

14.X MVC mit Struts-Framework

- Action.perform – Methode

```
public ActionForward perform(  
    ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response)  
    throws IOException,  
    ServletException  
{  
    . . .  
    return  
        (mapping.findForward("success"));  
}
```

14.X MVC mit Struts-Framework

Using Struts

- Imports

```
<%@ taglib uri="/WEB-INF/struts.tld"
      prefix="struts" %>
```

- Text

```
<struts:message key="join.title"/>
```

- Fehlerausgabe

```
<form:errors/>
```

- HTML Form

```
<form:form action="join.do" focus="email" >
```

- HTML `<input>` ==> `<form:text/>`.
- HTML `<submit>` ==> `<form:submit/>`.

14.X MVC mit Struts-Framework

