Seminar „DB-Aspekte des E-Commerce“
Schwerpunkt: Techniken

Servlets und JavaServer Pages

Boris Stumm

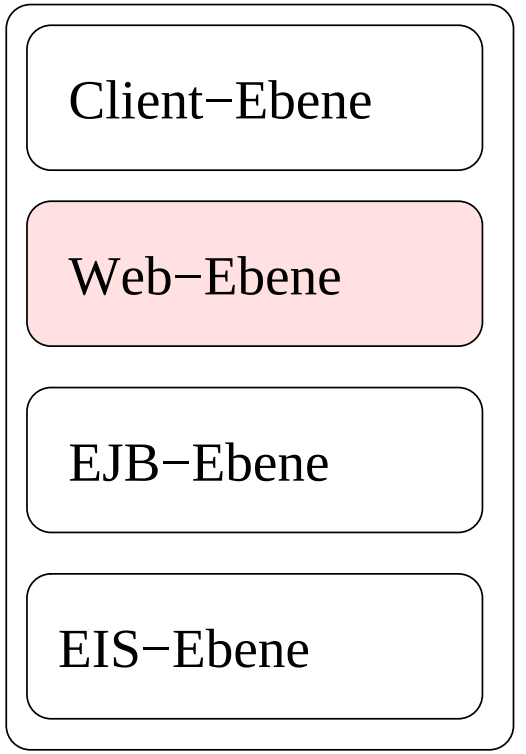
AG Datenbanken und Informationssysteme

Inhalt

- Einführung
- Technologie der Servlets und JavaServer Pages
  - Umgebung und Begriffe
  - Servlets – Aufbau, Funktionsweise, Eigenschaften
  - JavaServer Pages – Aufbau, Funktionsweise, Ziele
  - Vergleich der beiden Web-Komponenten-Technologien
- Architekturmodelle
  - Überblick
  - MVC-Pattern und andere Design-Patterns
  - Web-zentrierte Modelle
  - EJB-zentrierte Modelle

Einführung

- Einordung in das J2EE-Framework
- Allgemeine Problemstellungen
  - Dynamische Erzeugung von Inhalten
  - Thin Clients
  - Trennung von Präsentation und Logik
- Von J2EE gebotene Möglichkeiten
  - Servlets
  - JavaServer Pages
  - (Enterprise JavaBeans)



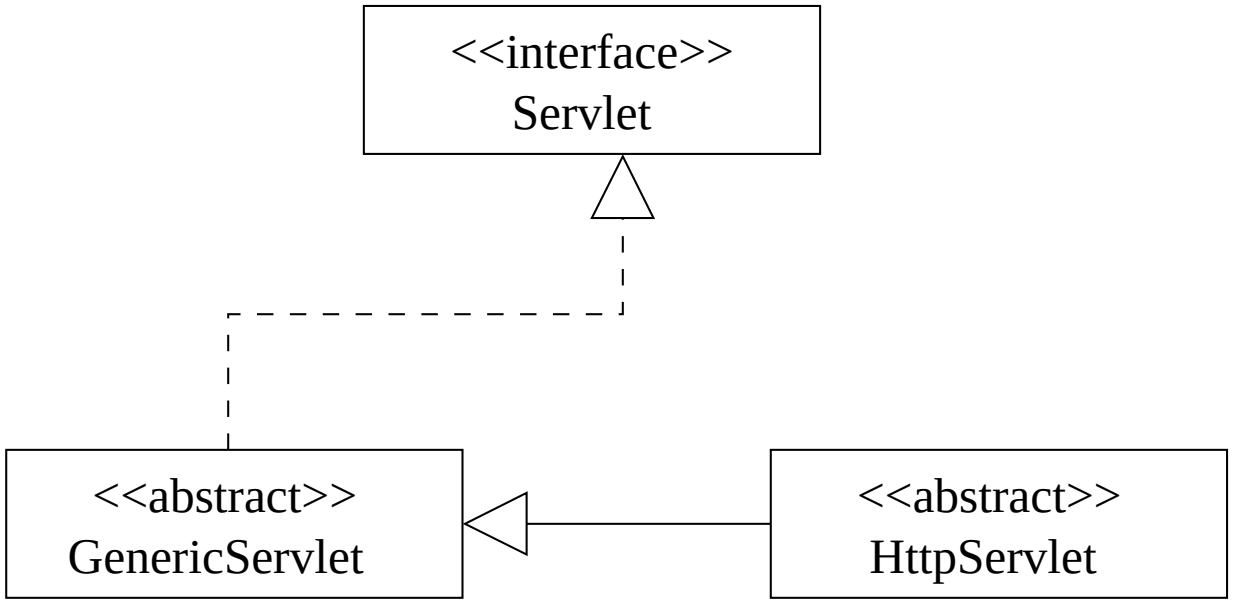
Servlets und JSP – Allgemeines

- Begriffe:
  - Web-Komponenten
  - Web-Applikationen
  - Web-Container
- Request/Response-Modell
- Servlets als Servererweiterung
- Sicherheit

Servlets (1) – API

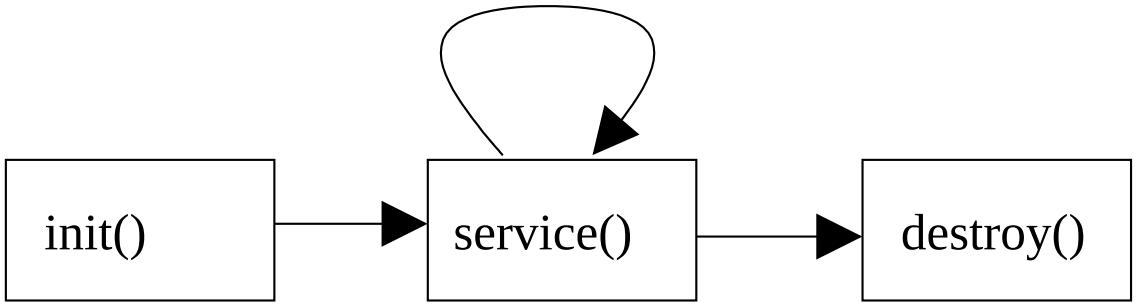
Das Servlet-API stellt verschiedene Interfaces und Klassen bereit

- `Servlet`
- `GenericServlet`
- `HttpServlet`



Servlets (2) – Eigenschaften

- Lebenszyklus
  - `init()`
  - `service()`
  - `destroy()`



- HTTP-Spezifische Servlets
- Sitzungsverwaltung
- Kommunikation mit der Umgebung und anderen Komponenten

Servlets (3) – Beispiel

```
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
                            throws ServletException, IOException {
        HttpSession ses = req.getSession();
        RequestDispatcher disp = req.getRequestDispatcher();

        if (ses.getAttribute("name")==null) {
            disp.forward(resp.encodeURL("/login");
        }
        else {
            // ...
            resp.getWriter().println("Hallo "+ses.getAttribute("name"));
            disp.include("/template/banner.html");
        }
    }
}
```

Servlets (4) – Zugriffsebenen

- Application Scope
  - `ServletContext`
  - Externe Resourcen wie Datenbanken
- Session Scope
  - `HttpSession`, erreichbar über `request.getSession()`
- Page Scope
  - Klassenvariablen
- Request Scope
  - Lokale Variablen
  - `HttpServletRequest`

Servlets (5) – Bemerkungen

- Performanz
  - Einmaliges Laden und Starten
  - Laufen in leichtgewichtigen Prozessen
  - Servlets vs. CGI
- Unkomfortable Behandlung der Ausgabe
- Vermischung von Präsentation und Logik

JavaServer Pages (1) – Motivation

- Trennung der Präsentation von der Logik
- Einbettung von Java-Kode in HTML
- JavaServer Pages (JSP) können von Webdesignern ohne größere Programmiersprachenkenntisse entworfen werden.
- Erweiterung der Servlet-Technologie

JavaServer Pages (2) – Elemente

- Statischer Text
  ```
  <H1>Welcome</H1>
  ```
- JSP-Skriptelemente
  - Ausdrücke
    ```
    <%= clock.getDayOfMonth() %>
    ```
  - Skripte
    ```
    <% } else { %>
    ```
  - Deklarationen
    ```
    <%! void doSomething() { doNothing(); } %>
    ```
- Direktiven
  ```
  <%@ include file="copyright.html" %>
  ```
- Aktionen
  ```
  <jsp:useBean id="clock"
               class="calendar.jspCalendar" />
  ```
- Implizite Objekte
  ```
  <% x = session.getAttribute("xValue") %>
  ```

JavaServer Pages (3) – Beispiel

```
<HTML>
<%@ page language="java" imports="com.wombat.JSP.*" %>
<jsp:useBean id="clock" class="calendar.jspCalendar" />

<H1>Welcome</H1>
<P>Today is </P>
<UL>
    <LI>Day: <%= clock.getDayOfMonth() %>
    <LI>Year: <%= clock.getYear() %>
</UL>
<%  Calendar c = Calendar.getInstance();
    if (c.get(Calendar.AM_PM) == Calendar.AM) { %>
        Good Morning
<% } else { %>
        Good Afternoon
<% } %>
<%@ include file="copyright.html" %>
</HTML>
```

JavaServer Pages (4) – Aktionen (Tags)

- Erweiterung der JSP-Sprache durch benutzerdefinierte Tags
- Erzeugt von Softwareentwicklern
- Benutzt von Webdesignern
- Wiederverwendung durch *Tag-Libraries*

JavaServer Pages (5) – JavaBeans

Beans als „The Only Component Architecture for Java Technology“

Bei JSP:

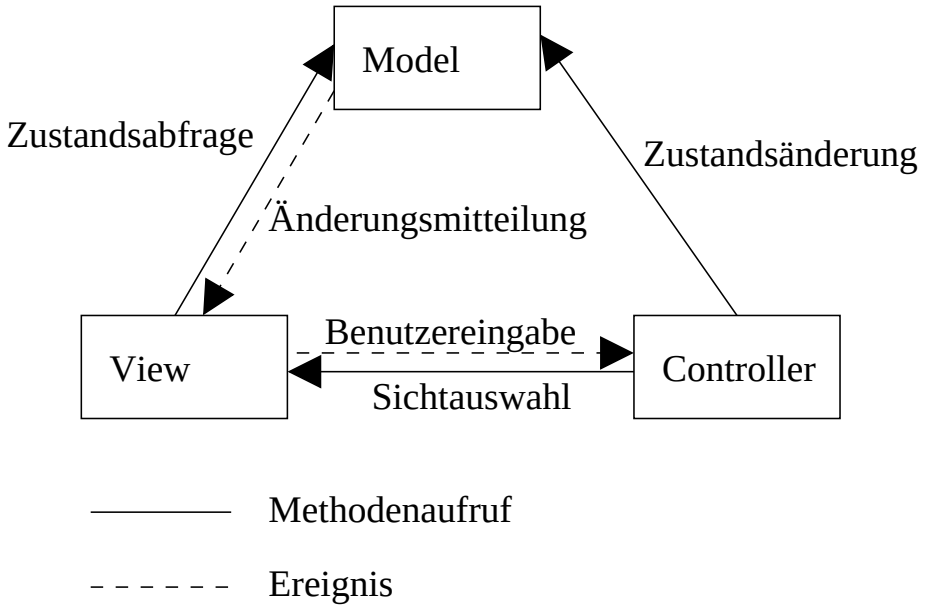
- Direkte Unterstützung durch Aktionen
- Design-Konventionen unterstützen Wiederverwendbarkeit
- JavaBeans mit Entwicklungstools bearbeitbar

Vergleich von Servlets und JSP

- JavaServer Pages
  - JSP als Erweiterung von Servlets ermöglichen neue Entwicklungsmethoden
  - Trennung von Präsentation und Logik durch JSP
- Servlets
  - Benutzung als Front Component
  - Dynamische Erzeugung von Binärdaten

Architekturmodelle – Überblick

- MVC als grundlegendes Pattern
  - Trennt Präsentation und Logik
  - Minimiert Kopplung der Komponenten
  - Ein Modell, mehrere Sichten
- Andere Patterns
  - Page-by-Page-Iterator
  - Session Facade



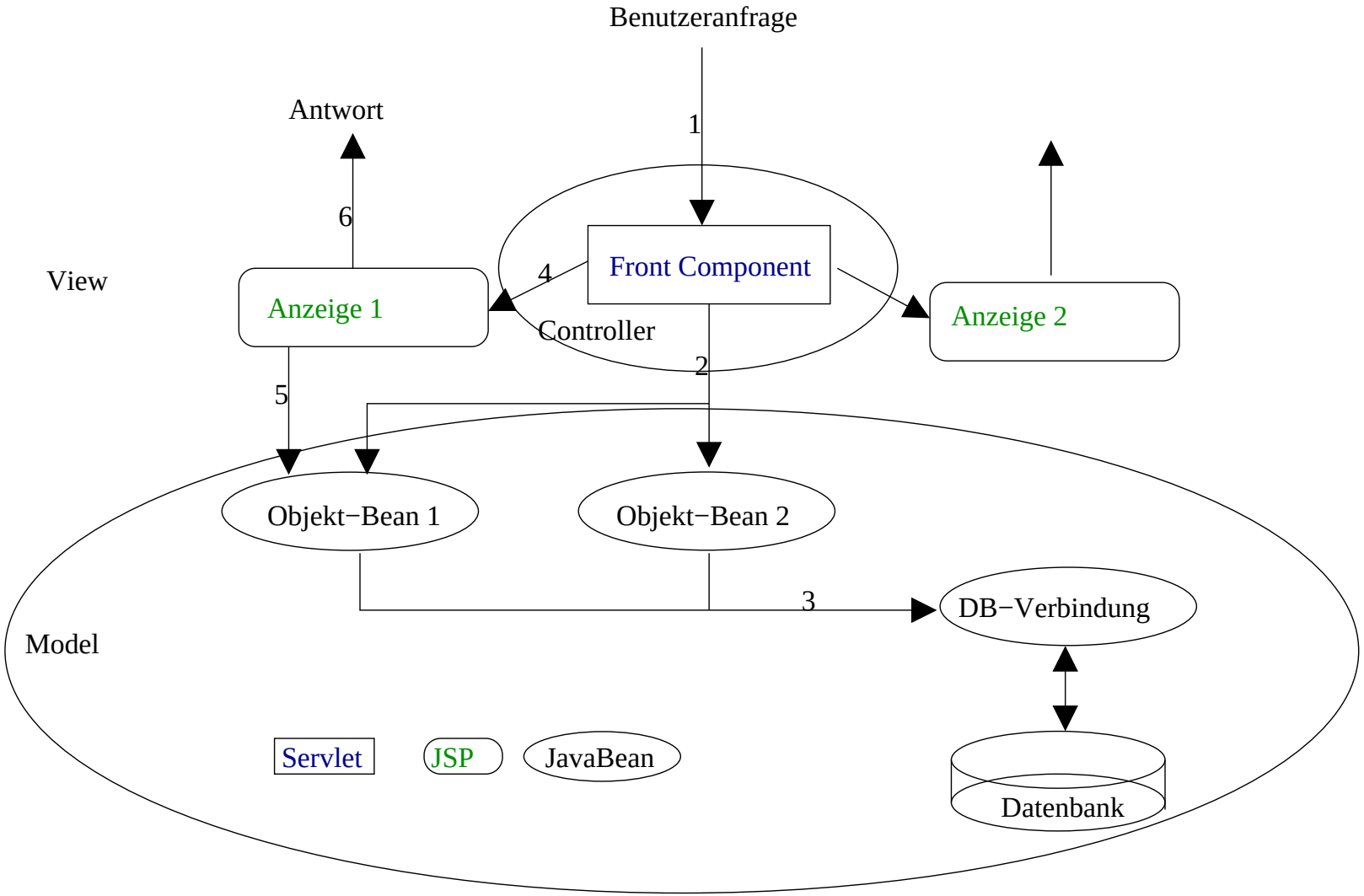
Entwurfsmöglichkeiten für Web-Basierte Applikationen

Je nach Anwendungs-Komplexität kommen verschiedene Modelle zum Einsatz

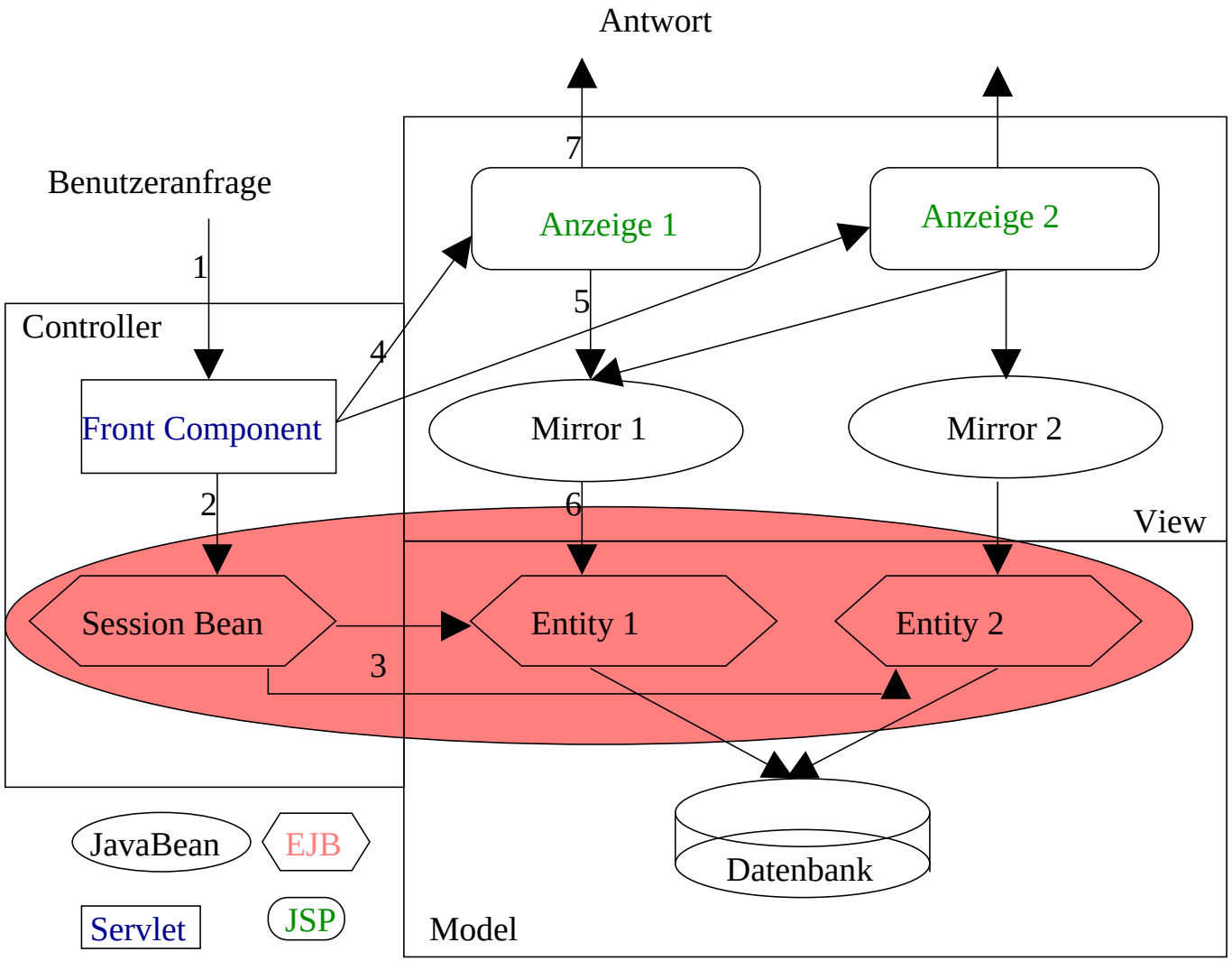
- Web-Zentriert
  - Statische HTML-Seiten
  - Einfache Servlets und JSPs
  - Zusätzliche Nutzung von JavaBeans und benutzerdefinierte Aktionen
- EJB-Zentriert
  - Nutzung von Enterprise JavaBeans

$\Longrightarrow$ Welche Möglichkeit soll im konkreten Fall angewandt werden?

Web-Zentrierte Applikationen



EJB-Zentrierte Applikationen



Zusammenfassung

- Problem der Erzeugung dynamischer Inhalte
- Webclients für Applikationen
- Servlets als Lösung
- Neue Möglichkeiten durch Benutzung von JSP
- Trennung von Logik und Präsentation durch JSP
- Architekturen:
  - Web-Zentriert
  - EJB-Zentriert