Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Application Server und Continuous Integration

Outline

- Einleitung Application Server
- Java EE
- Enterprise Applikationen vs. Web Applikationen
- Web Application Life Cycle
- Servlets
- JavaServer Pages
- verschiedene Application Server
- JSP/Servlet Demo
- Continuous Integration
- Jenkins
- Jenkins Demo

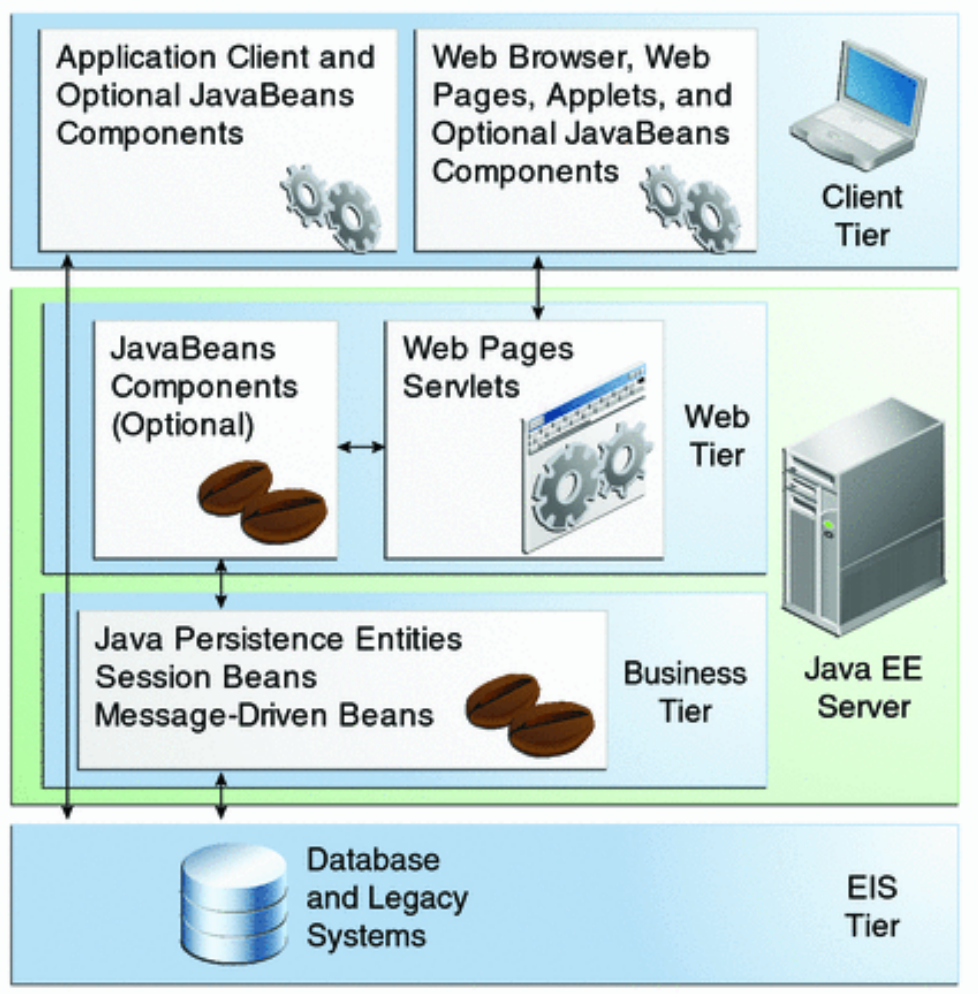
Application Server

- **ursprünglich**
  - Ausführungsumgebung für beliebige Applikationen
  - z.B. .NET, Java EE
- **heute**
  - Ausführungsumgebung für Web-Applikationen
  - normalerweise Synonym für Java Application Server

Java EE

- Platform mit verschiedenen Technologien
- reduziert Entwicklungskosten und Komplexität
- Standard für Java Server-Applikationen
- erweitert Java SE

Java EE


Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
JavaBeans Components (Optional)
Web Pages Servlets
Web Tier
Java Persistence Entities
Session Beans
Message-Driven Beans
Business Tier
Java EE Server
Database and Legacy Systems
EIS Tier

Aufgaben von Java EE

- Sicherheit
- Transaktionsmanagement
- Namens- und Verzeichnisdienste
- Kommunikation zwischen Java-EE Komponenten
- Persistenzdienste
- Deployment
- Kapselung von Ressourcen

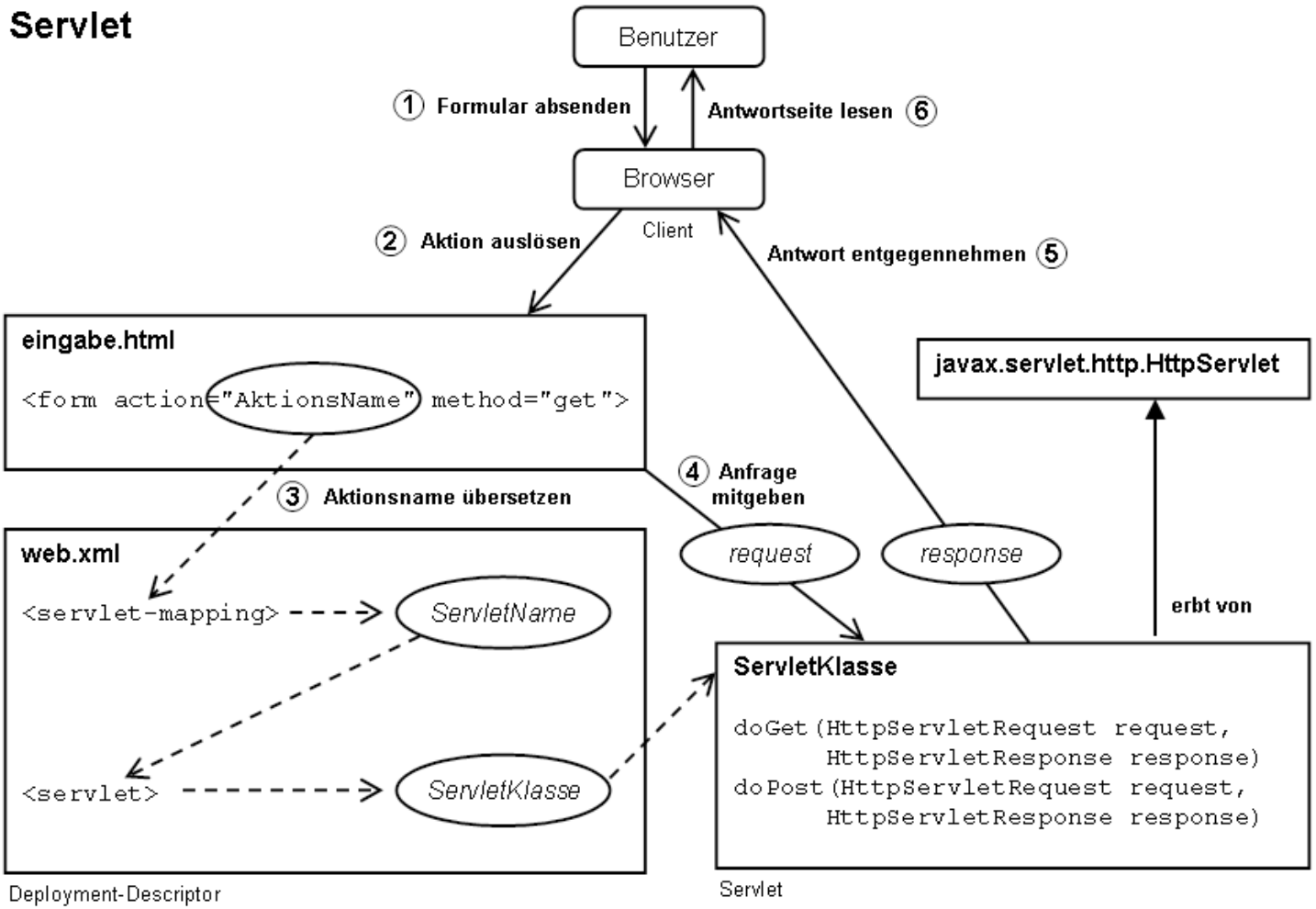
Web Application Lebenszyklus

1. Web-Komponente entwickeln
2. Deployment Descriptor für Web Anwendung schreiben(web.xml)
3. Komponenten der Web-Anwendung kompilieren
4. optional Paket erstellen (.war Datei)
5. Anwendung in einen Container legen
6. per URL drauf zugreifen

Servlet

- *servlet* = **serv**er + app**let**
- Java-Klassen, die Anfragen von Clients verarbeiten
- implementieren Schnittstelle `javax.servlet.Servlet`
- erbt normalerweise von `javax.servlet.http.HttpServlet`
- fester Bestandteil aller Java-EE Application Server
- Instanzen werden in *web containern* erzeugt
- implementieren doGet, doPost
- bei JSP: _jspInit(), _jspService(), _jspDestroy()

Servlet


Servlet
Benutzer
1 Formular absenden
Antwortseite lesen 6
Browser
Client
2 Aktion auslösen
Antwort entgegennehmen 5
eingabe.html
<form action="AktionsName" method="get">
javax.servlet.http.HttpServlet
3 Aktionsname übersetzen
4 Anfrage mitgeben
web.xml
request
response
<servlet-mapping>
ServletName
erbt von
ServletKlasse
doGet(HttpServletRequest request,
HttpServletResponse response)
doPost(HttpServletRequest request,
HttpServletResponse response)
<servlet>
ServletKlasse
Deployment-Descriptor
Servlet

Servlet Beispiel


```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class HelloWorld extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("<html>\n" +
                "<head><title>Hello World</title></head>\n" +
                "<body>\n" +
                "<h1>Hello, world!</h1>\n" +
                "</body></html>");
  }
}
```

Servlet Descriptor (web.xml)


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
   PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
   "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
 <display-name>Your project name</display-name>
 <description>
   Your servlets.
 </description>
 <servlet>
   <servlet-name>YourServlet</servlet-name>
   <servlet-class>YourServletClass</servlet-class>
 </servlet>
 <servlet-mapping>
   <servlet-name>YourServlet</servlet-name>
   <url-pattern>/YourServlet</url-pattern>
 </servlet-mapping>
</web-app>
```

JavaServer Pages (JSP)

- Web-Programmiersprache von SUN
- basiert auf JHTML
- Integration von Java-Code über Scriptlets
- JSP-Engine kompiliert Code zu *Servlet*
- konkurriert zu ASP, CGI, PHP

```
<html>
<head><title>First Example</title></head>
<body>
<h3>Hello World-JSP</h3>

Your browser is: <%= request.getHeader("User-Agent") %><br />
Your IP address is: <%= request.getRemoteAddr() %><br />

</body>
</html>
```

Enterprise Application vs. Web Application

Enterprise Application	Web Application
• EAR Dateien • Java Servlets • JavaServer Pages • JavaServer Faces • Java Authentication und Authorization Service (JAAS) • J2EE Connector Architecture • JavaBeans Activation Framework (JAF) • JavaMail • Java Message Server (JMS) • Java Persistence API (JPA) • Java Transaction API (JTA) • Java Management Extensions API (JMX) • Java API for XML Processing (JAXP) • Java API for XML-based RPC (JAX-RPC) • Java Architecture for XML Binding (JAXB) • SOAP with attachments API for Java (SAAJ) • Java Database Connectivity (JDBC) Framework	• WAR Dateien • Java Servlets • JavaServer Pages • JavaServer Faces • Java Database Connectivity (JDBC) Framework

jetty://

- JSP/Servlet Container und Webserver
- eigentlich kein Application Server (Java EE nicht komplett)
- open source
- verwendet von Application Servern: JOnAs und Geronimo
- benutzt JSP-Compiler Jasper von Apache Tomcat
- verwendet von Google Web Toolkit seit Version 1.6

Tomcat (1)

- Servlet/JSP Container und Webserver
- nur für Web-Anwendungen gedacht
- populärster Server für Java-Web-Anwendungen
- früher Apache Jakarta Projekt
- Open Source

Tomcat (2)

- auch standalone Web Server
- in ¼ aller Produktionssysteme eingesetzt
- komplett in Java geschrieben
- JSP Compiler: Jasper
- keine wesentlichen Unterschiede zu Jetty

Demo

Glassfish

- enstanden aus Sun Java System Application Server (SJSA)
- Referenz-Implementierung von Java-EE Spezifikation
- hat als erste Java-EE Implementierung

- älteste volle Java-EE Implementierung
- gute Reputation
- Teil von RedHats Middleware
- Update auf neuste Java-EE immer später als Glassfish
- langsamer als Glassfish

Continuous Integration

- ständiges Bauen und Testen einer Anwendung
- Überprüfung neuer Versionen in VCS
- Entwickler über evtl. Probleme informieren
- schnelle Aufspürung von Fehlern
- dadurch: konstante Verfügbarkeit eines lauffähigen Systems
- Voraussetzung: TDD

Jenkins

- ursprünglich von SUN Microsystems: *Hudson*
- in Java geschrieben
- läuft in beliebigem Servlet-Containern
- in minimalem Servlet-Container *Winstone* ausgeliefert
- unterstützt verschiedene Build-Tools, wie *ant* oder *maven*
- unterstützte VCS: CVS, Subversion
- unterstützte Testumgebungen: JUnit, Emma
- durch viele Plugins erweiterbar

Demo

Quellen

- http://download.oracle.com/javaee/6/tutorial/doc/
- http://tomcat.apache.org/
- http://jetty.codehaus.org/jetty/
- http://glassfish.java.net/
- http://jenkins-ci.org/
- http://jboss.org/