



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Enterprise Web Development (Java Enterprise Edition 6)



wissen.leben
WWU Münster

Enterprise Web Development
Dozent: Arne Scheffer

Inhalt und Organisation der Lehrveranstaltung

- In dieser Lehrveranstaltung wird die effiziente Entwicklung von Webanwendungen mit Hilfe fortgeschrittener Sprachmittel von Java vorgestellt.
- Die Lehrveranstaltung ist für Teilnehmer mit Programmierkenntnissen ausgelegt
 - Minimalanforderungen sind
 - die Kenntnis einer Programmiersprache (vorzugsweise Java, C#)
 - Grundkenntnisse in der Skriptsprache (X)HTML
 - Grundkenntnisse in SQL
- Die Teilnehmer müssen sich anmelden:
<https://www.uni-muenster.de/ZIV/zivlehre.html>

Vorlesungsbesuch und Scheinkriterien

- Die Lehrveranstaltung besteht im Wesentlichen aus Vorlesungen
 - Diese finden jeweils mittwochs zwischen 10 und 12 Uhr im M4 statt.
 - Genaue Anfangszeit → Auditoriumsvotum
 - das Vorgehen wird überwiegend live demonstriert und ist zur Wiederholung/Übung weitgehend in den Unterlagen dokumentiert
 - kann so einfach nachvollzogen werden
- Teilnahmebescheinigung
 - Wenn mindestens 75% der Vorlesungen besucht werden,
 - kann die Teilnahme bescheinigt werden (ohne Zusatz „erfolgreich“)
- Leistungsschein/Punkte
 - Durch zusätzliche Teilnahme an einer Prüfung
 - können darüber hinaus 3 Punkte im Rahmen der allgemeinen Studien erworben werden (Ausnahmen in Informatik-Studiengängen beachten!)
 - kann die erfolgreiche Teilnahme bescheinigt werden (benotet)

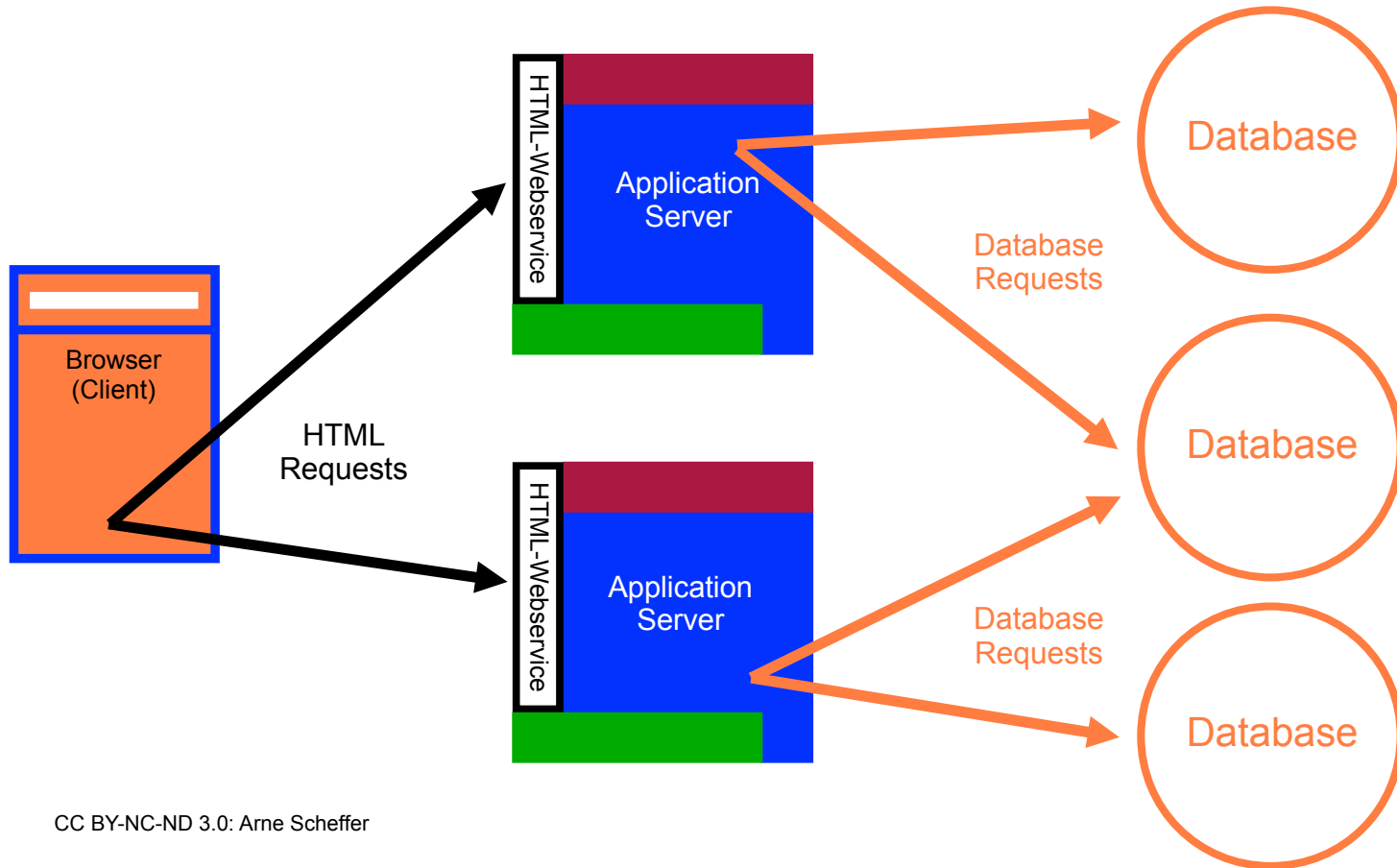
Titel der Lehrveranstaltung

- „Enterprise Web Development (JEE 6)“
- sehr umfangreiches Framework
 - allein die Neuerungen gegenüber Version 5 würden eine Vorlesung füllen
 - besser in einem BLOG nachlesen
 - die Gesamtübersicht über alle Features würde eine Vorlesung füllen
 - Komplexität und Zeitrahmen verhindern eine anschauliche Darstellung
- also nähere Spezifikation/Fokus nötig:
- „Effiziente Entwicklung von Webanwendungen“ (s. auch Vorlesungskommentar)
 - Verwendung des JEE6 Web Profile
 - bedarfsweise Ergänzung um weitere Bausteine aus dem JEE-Rahmenwerk
- Warum dieser Fokus ?
 - Dozent arbeitet im Umfeld Webanwendungen
 - JEE6 bietet hier Neuerungen die Wartbarkeit der Anwendungen betreffend
 - Mehrwert für Dauereinsatz und Entwicklung

Was wollen wir erreichen ?

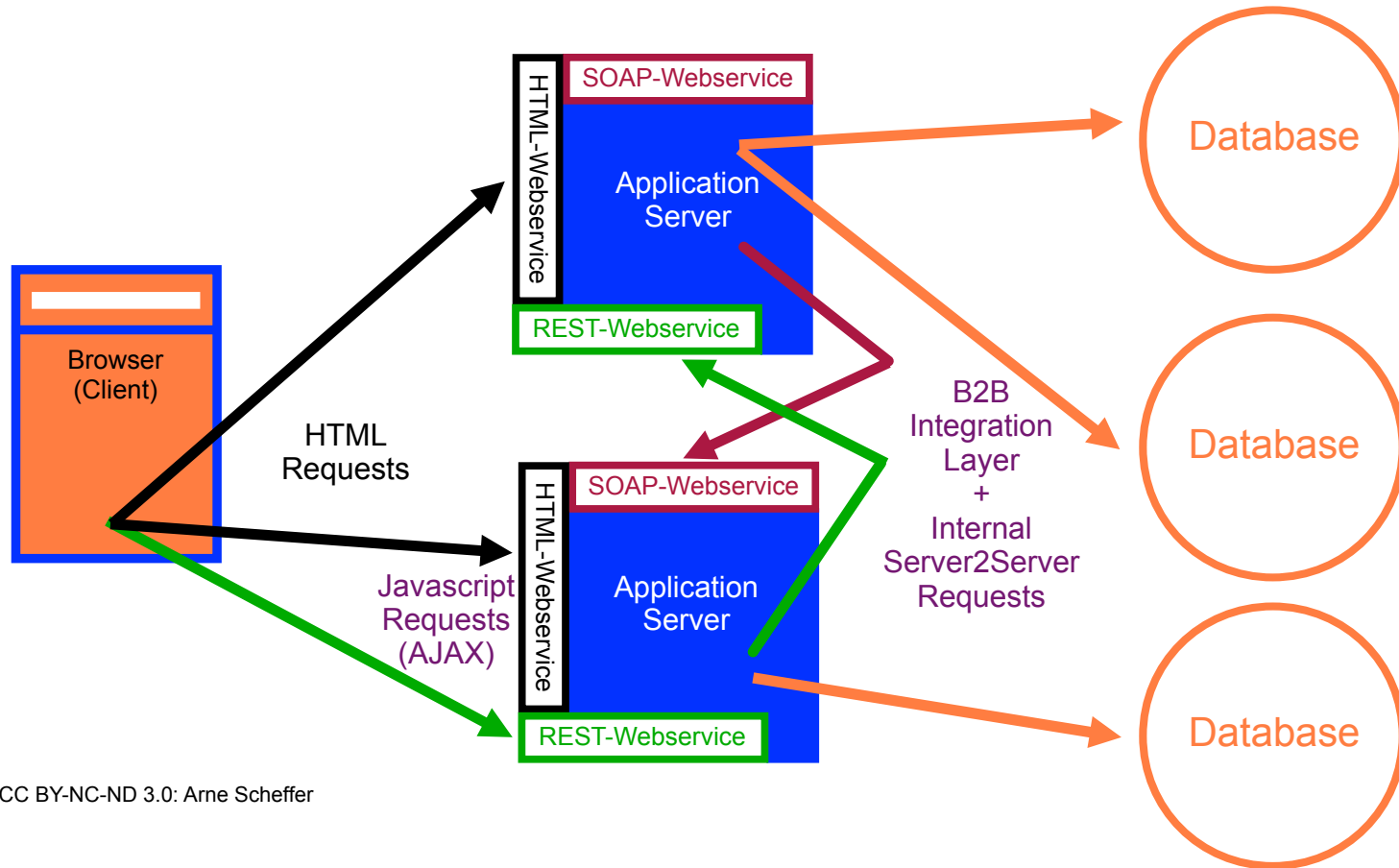
- Ziele eines Web-Frameworks wie JEE6 Web Profile
 - Trennung von Sicht, Business-Logik und Datenmodell/basis (MVC, MVP)
 - entspricht: Erleichterung der Wartbarkeit durch Modularisierung in Bausteine
 - getrennte Sicht
 - Templating-Engines wie bei einem WCMS
 - Tag-Libraries z.B. für Buttons oder Drop-Down-Menüs
 - getrennte Business-Logik, Datenbasis
 - Austauschbarkeit/Erweiterbarkeit/Trivialisierbarkeit
 - Abstraktion sich wiederholender Abläufe einer Webanwendung in Request-Zyklen
 - Bereitstellung vorgefertigter Module z.B.
 - zur Authentifizierung
 - für Prozess-“Automaten“
 - Datenbankbindung auf Konfigurationsbasis statt Implementierungsbasis

Einführung: Application Server / Webserver



CC BY-NC-ND 3.0: Arne Scheffer

Ausblick: Webservices, Application Server ↔ Webserver



CC BY-NC-ND 3.0: Arne Scheffer



Themenübersicht für die Vorlesung

- MVC-Konzept
- Rapid CRUD
- Java Server Faces
 - Facelets
- Kapselung
 - CDI
- ORM
 - Entities
- Webservices
 - SOAP
 - REST
- AJAX
- Bean-Typen
- ...

„Geht das nicht auch einfach mit PHP ?“

- die Frage zeigt mangelndes Verständnis
- ungefähr vergleichbar mit
 - „Kann ich PHP-Programme nicht auch direkt in Maschinensprache schreiben?“
- keine Frage der Programmiersprache, sondern abstrahierender Überbau
- Alternativen: Spring (Java), Struts (Java), php-Cake, Ruby on Rails
- Web-Framework bringt für jedwelche Programmiersprache
 - geringere Komplexität bei der Entwicklung einer typischen Webanwendung
 - potentiellen Mehraufwand bei der Entwicklung exotischer Anwendungen
 - Aufwand bei der Schulung der Umgebung und der Werkzeuge
- immer abwägen, ob wirklich die Waffe der Wahl, denn
 - das gute alte AMP-Skript hat im Kleinen eine wesentlich geringere Komplexität
 - wird nach Jahren der Aufrüstung um Features zum Wartungsdisaster

Verwendete Mittel

- benötigt werden
 - JEE6-Server
 - IDE
 - Plattformunabhängigkeit
- verwendet für die Vorlesung
 - Bundle aus Netbeans und Glassfish V3
- einfach auf der Netbeans-Seite für das gewünschte Betriebssystem runterladen
- Nach der Installation:
 - Achtung !
 - mehrere Server-Dienste installiert (z.B. Tomcat-Web-Server)
 - => Ports für die Außenwelt schließen, z.B. per Firewall
 - für die Vorlesung reicht die lokale Verwendung

Im JEE6 Web Profile verwendete Bausteine/Begriffe

Diese für den Einstieg unerlässlichen Begriffe werden nachfolgend erklärt

- Java Server Faces (JSF)
 - JSF-Component-Tree
 - Facelets
 - Beans, Entities
 - Persistence-Unit
 - Lifecycle-Unit
 - ORM
-
- Top-Down-Vorgehen für schnellen Einstieg
 - Black-Boxing, Gray-Boxing
 - mangelnde Kenntnis der Details ersetzt durch automatisierte Erstellung
 - später: Tiefenverständnis steigern
 - später: Portfolio vergrößern

Web Page

(based on HTML tag tree)

```
graph TD; A(( )) --- B(( )); A --- C(( )); B --- D(( )); B --- E(( )); C --- F(( )); E --- G(( )); E --- H(( )); G --- I(( )); G --- J(( )); H --- K(( )); H --- L(( ));
```

</html>

```
<html>
  <head><title>HTML-Seite</title></head>
  <body>
    <h1>HTML-Seite</h1>
    <p>Dies ist eine HTML-Seite.</p>
  </body>
</html>
```

- `<html></html>`
 - `<head></head>`
 - `<title></title>`
 - HTML-Seite
 - `<body></body>`
 - `<h1></h1>`
 - HTML-Seite
 - `<p></p>`
 - Dies ist eine Beispielseite.

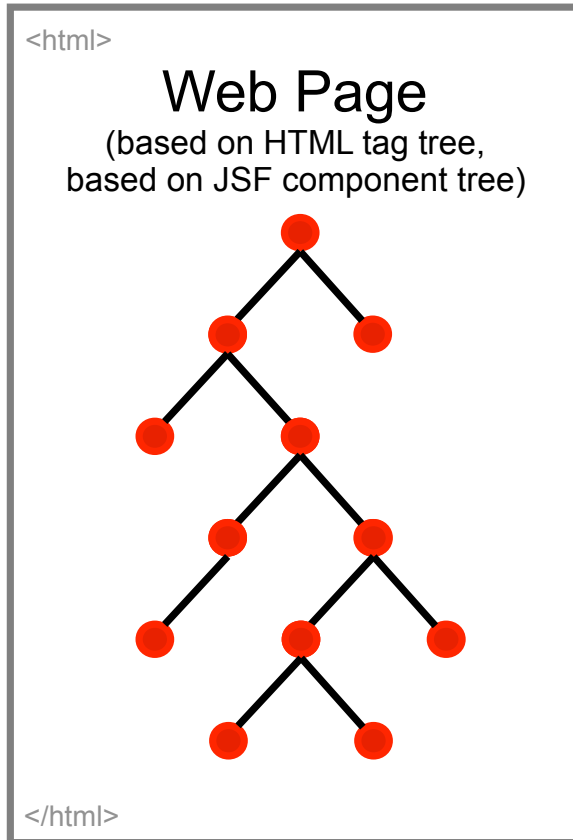
Einführung in Java Server Faces: Tag-Libraries

- Verwendung vorgefertigter View-Komponenten
 - Buttons
 - Eingabefelder
 - Ausgabefelder
 - Listen/Tabellen
 - Select-Boxen etc.
 - selbstdefinierte Komponenten
- mit Variablen parametrisierbar
- im Header der JSF-Seite ist die zu verwendende Library anzugeben

Komponenten-Einbau in die HTML: JSF-Component-Tree

- „Viewhandler“ erforderlich, um Tags und Libraries in die HTML-Seite einzubauen
 - Parser, der eine Skriptsprache versteht (View Declaration Language)
 - eingebaute JSF-Komponenten werden
 - erkannt und
 - an das JSF-Framework weitergereicht
 - der JSF-Component-Tree wird erzeugt/aktualisiert
 - JSF liefert auf dessen Basis (später) passendes HTML zurück
- Verwendung von Facelets als Viewhandler-Technologie
 - XML-konform
 - Templating
 - passt zu den JSF-Lifecycle-Phasen
- aufgrund der obigen Vorteile
 - sind Facelets mittlerweile Standard
 - werden wir die Alternative Java Server Pages hier nicht behandeln

Einführung in Java Server Faces: Facelets-Struktur



```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>HTML-Seite</title>
  </h:head>
  <h:body>
    <h1>HTML-Seite</h1>
    <p>Dies ist eine HTML-Seite</p>
    <p>Sie wurde
    <h:outputText value="#{aufrufe.anzahl}"></h:outputText>
    mal aufgerufen</p>
  </h:body>
</html>
```

Web Page

(based on HTML tag tree,
based on JSF component tree)

```
graph TD; A(( )) --- B(( )); A --- C(( )); B --- D(( )); B --- E(( )); C --- F(( )); D --- G(( )); D --- H(( )); E --- I(( )); E --- J(( )); F --- K(( )); F --- L(( )); H --- M(( )); H --- N(( ))
```

</html>

- ...
 - h:head
 - <title></title>
 - HTML-Seite
 - h:body
 - <h1></h1>
 - HTML-Seite
 - <p></p>
 - Dies ist eine HTML-Seite.
 - <p></p>
 - Sie wurde
 - h:outputText value="#{aufrufe.anzahl}"
 - mal aufgerufen

Dozent: Arne Scheffer

Datum: 17.10.2012

Einführung in Java Server Faces: JSF-Lifecycle

vereinfacht:

- Facelets-Seite
wird geparkt
- JSF-Component-Tree
JSF rendert
- HTML-Seite
Request
- Facelets-Seite
wird geparkt
- JSF-Component-Tree
JSF rendert
- HTML-Seite
Request
usw. ..

- JSF-Lifecycle

Eine JSF-Anwendung durchläuft mit jedem Request (erneut)
bis zu sechs *Phasen*:

- *Restore View* („Sicht wiederherstellen“)
- *Apply Request Values* („Anforderungsparameter anwenden“)
- *Process Validations* („Validierung ausführen“)
- *Update Model Values* („Modell aktualisieren“)
- *Invoke Application* („Anwendung aufrufen“)
- *Render Response* („Antwort wiedergeben“)

Daten-Persistenz in JEE

- Persistiert man Daten nicht, sind sie nach Anwendungsende verloren
- Daten werden von einer Anwendung zumeist in einer relationalen (SQL-)Datenbank gespeichert. (Ausnahmen: Files, No-SQL-Datenbanken)

Probleme:

- Das Streuen von SQL-Anweisungen im Code macht Anwendungen schlecht wartbar (Modell-Erweiterungen/-Änderungen)
- Bruch mit der objektorientierten Java-Welt

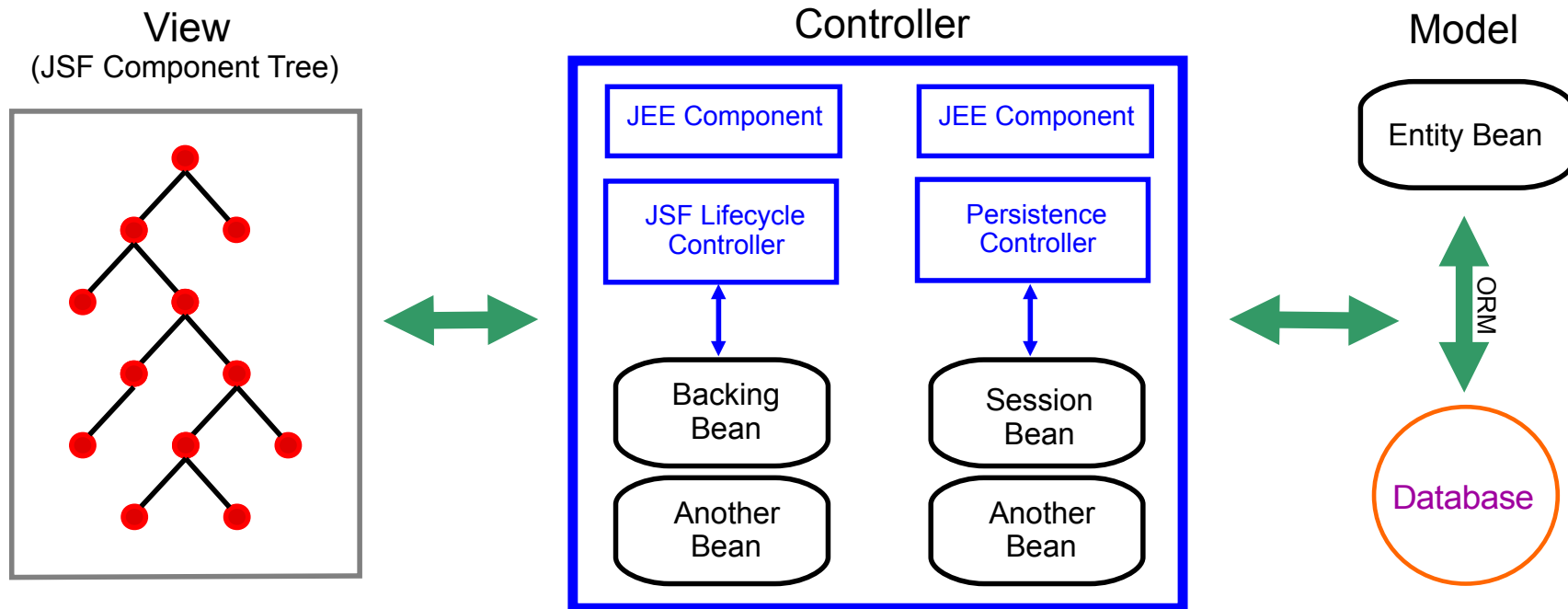
Lösung:

- automatisches Mapping von Tabellenzeilen auf einfache Java-Objekt-Instanzen
 - „O(bject)R(elational)M(apping)“ durch eine „Persistence-Unit“
 - Spaltendefinition der Tabellen durch Java-Objekte mit Spalten-Attributen
 - annotiertes „P(lain)O(ld)J(ava)O(bject)“ --> „Entity“

Model-View-Controller-Konzept

- strikte Gliederung der Code-Bestandteile einer (Web-)Anwendung in
 - Datenmodell
 - Sicht (View)
 - Businesslogik (Controller)
- Vorteile:
 - Trennung der Sicht vom Rest macht die GUI einfacher änder-/erweiterbar
 - Teile der Businesslogik können
 - automatisiert erstellt werden
 - mittels vorgefertigter Plugins abgewickelt werden
 - Unterstützung durch das Framework: Persistenz-Unit, ORM
 - Datenmodell konfigurieren statt implementieren
 - Delegation an Spezialisten möglich

MVC-Konzept im JEE-Framework



CC BY-NC-ND 3.0: Arne Scheffer

- die Business-Logik („Controller“) steuert frameworkseitig u.a.
 - den JSF-Lifecycle
 - das Persistieren der Daten

Demonstration der Erstellung einer Webanwendung

- Linkliste mittels „Rapid CRUD“
 - Basis: SQL-Tabellenstruktur

```
DROP TABLE link;  
CREATE TABLE link(  
  link_id INT NOT NULL,  
  link_name VARCHAR(30) NOT NULL,  
  link_url VARCHAR(150) NOT NULL,  
  primary key (link_id)  );  
INSERT INTO link VALUES (1,'Permail','https://permail.uni-muenster.de');
```
 - Einlesen in die Datenbank
 - automatische Generierung einer Entity mittels Netbeans
 - automatische Generierung von JSF-Seiten mittels Netbeans
 - Customizing zur Linkliste

Literatur

- Es gibt eine unüberschaubare Vielfalt von Büchern und Artikeln, die sich mit JEE beschäftigen, daher vorab nur eine kleine Auswahl von Artikeln und ein Tip
- Internet:
 - Tutorials verlinkt in Netbeans
 - Sun/Oracle Video-Tutorials zu JEE6 (Arun Gupta)
 - Facelets fits JSF like a glove (Richard Hightower)
 - jsf2fu series (David Geary)
 - What's New in JSF 2 ? (Andy Schwartz)
- Bücher:
 - Empfehlung: Sehr viele Seiten machen nicht unbedingt ein für einen Anfänger besonders geeignetes Buch, es sei denn zur Referenz.