

# Web Frameworks



## Einführung

Jürg Luthiger / Stefan Meichtry  
University of Applied Sciences Northwestern Switzerland  
Institute for Mobile and Distributed Systems

# Einführung

- Lernziele des Moduls “Web Frameworks”
- Inhalte
- Aufbau
  - Theorie zur Einführung eines neuen Lerninhaltes
  - Arbeitsblätter für praktische Vertiefung der Theorie
  - Übungen als Hausaufgaben zur Verfestigung des Lernstoffes
- Prüfungen

# Lernziele

**Die Studierenden kennen die wichtigsten Konzepte zur Entwicklung von Webapplikationen und können diese korrekt einsetzen.**

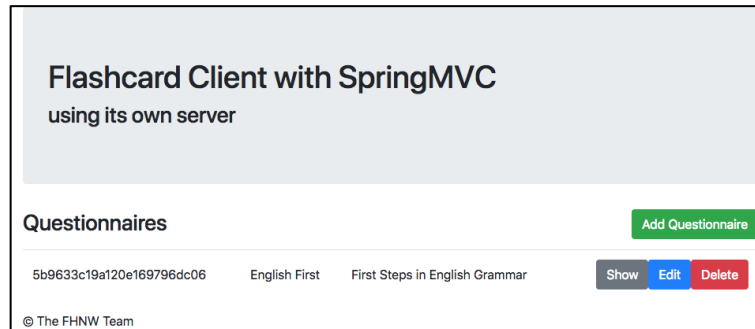
## Der Studierende

- kennt die typischen Herausforderungen einer modernen Webanwendungen und kann Lösungsansätze aufzeigen
- kann (auch neue) Technologien für den Bau von Webanwendungen einordnen und beurteilen
- kann die grundlegenden Konzepte einer Model-2 (MVC) Architektur erklären
- kennt die typischen Entwurfsmuster für Web Interfaces
- kennt clientseitige Technologien und entsprechende Frameworks und kann diese einsetzen

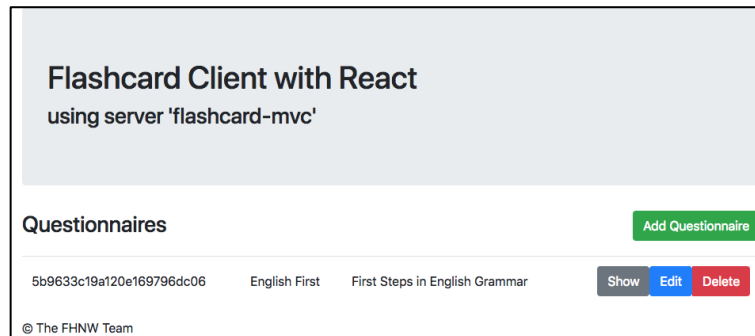
# Konzept

- Das Modul basiert auf 3 Teilen
  - Standard Web Applikationen
  - Single Page Application (SPA) mit REST-Service in Java
  - SPA mit REST-Service in JavaScript
- Das Lab "flashcard" dient als roter Faden über alle Teile
  - Analoges User Interface trotz Einsatz unterschiedlicher Technologien
- Beschränkung auf Java und JavaScript
  - Client: JavaScript
  - Server: Java und JavaScript
  - Datenbank: MongoDB

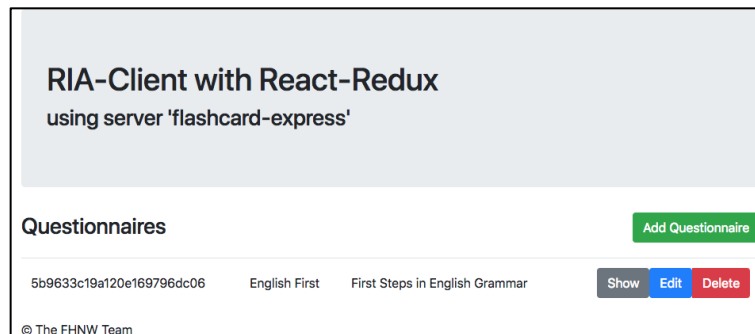
# Lab "flashcard": Simple CRUD-Applikation



SpringMVC

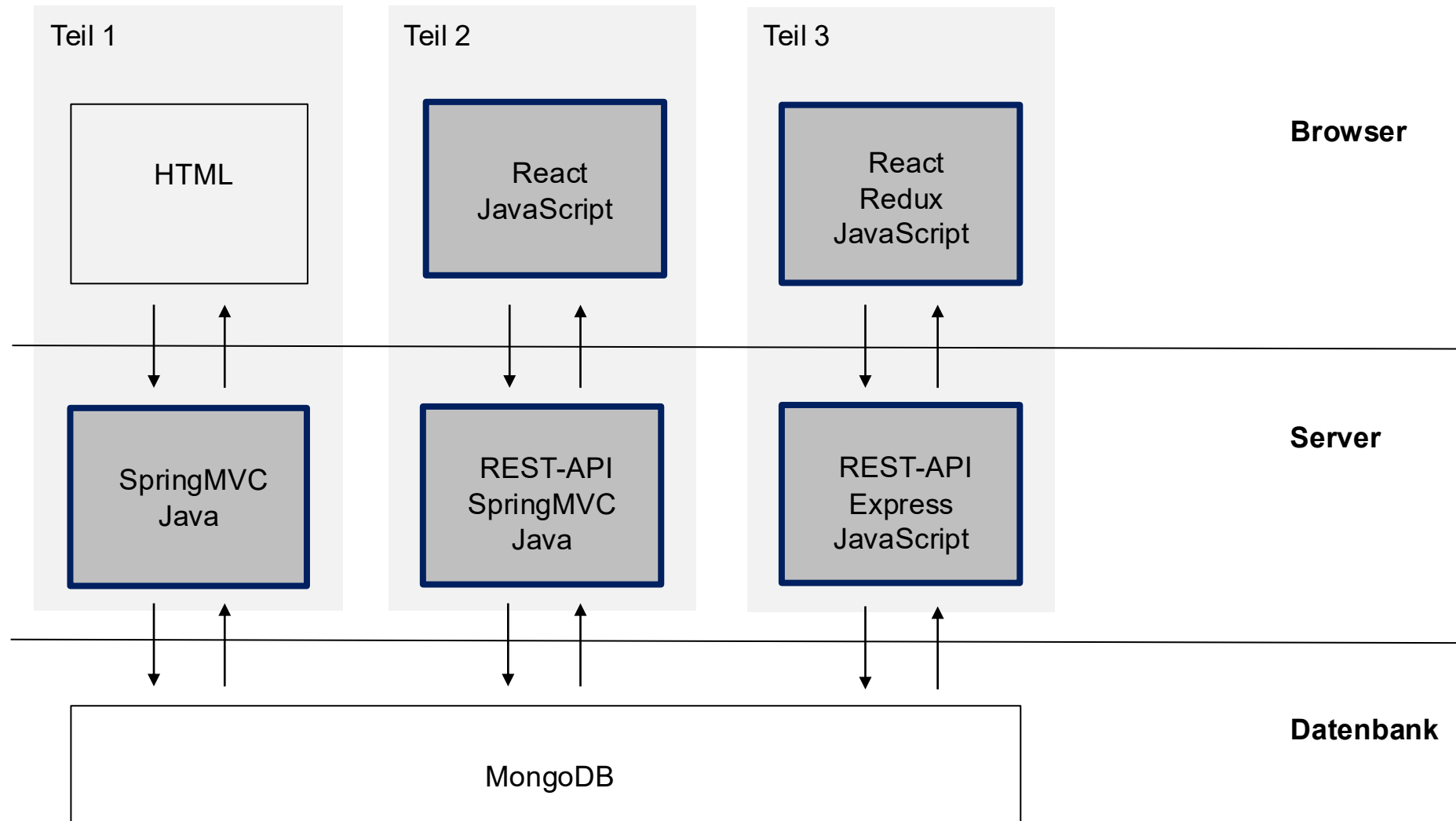


React & SpringMVC



React & Express

# Lab "flashcard": Setup



# Kursinhalt

## ■ Serverseitige Technologien

- Basiskomponenten der Java Servlet Technologie
- Strukturierung einer Web Applikation mit Spring MVC
  - Separierung der Controller Logik
  - Komponentenorientierte Views und Layout mit Thymeleaf
- Server mit JavaScript
  - Express als Web-Framework für Node.js

## ■ Clientseitige Technologien

- Single Page Application (SPA) mit React & State Manager
- Strukturierung einer Webapplikation
- Kommunikation zwischen Client und Server

# Unterrichtsmethode "Flipped Classroom"

- Der Anlass ist eine **Präsenzveranstaltung**.

**Aber**, die Theorie wird nicht vor Ort an der Hochschule vermittelt. Die Studierenden erarbeiten sich die Inhalte asynchron, ortsunabhängig, individuell, selbstgesteuert und im eigenen Lerntempo anhand von digitalen Lernmaterialien.

- Oft sind dies Videos, z.B. Screencasts; es können aber auch Podcasts, schriftliche Unterlagen, digitale Skripte und weitere Materialien zum Einsatz kommen.

**Sondern**, die wertvolle Präsenzzeit wird zur Vertiefung der Theorie genutzt, z.B. durch:

- Präsentation eines Fokusthemas.
- Diskussionen, Q&A.
- gemeinsame Aufgabenbearbeitung.

# Kursunterlagen

## Active Directory

- .../E2411\_Unterrichte\_Bachelor/E2411\_Unterrichte\_I/5ICa/webfr
  - Drehbuch HS25

## Gitlab

- Jede Lektion hat ein separates Gitlab-Projekt und beinhaltet oder verlinkt alle Unterlagen für die Lektion, so dass eine Vorbereitung@Home möglich ist:
  - Quellcode für Arbeitsblätter, inkl. Lösungen jeweils in einem separaten Ordner "complete"

## E-Mail

- Besondere Informationen während dem Semester erfolgen über E-Mail

# Prüfungen

## ■ Modulnote

- Prüfung 1 zu den Standard Technologien  
**tbd**
- Prüfung 2 zu den modernen Technologien  
**tbd**

**... fließen zu je 50% in die Modulnote ein!**

## ■ Organisation der Prüfungen:

- 30 Min theoretischer Teil  
**keine Hilfsmittel erlaubt**
- 60 Min praktischer Teil  
**schriftliche Hilfsmittel erlaubt, aber keine elektronischen Hilfsmittel**

# A Layered Architecture

z.B. im Java Ökosystem

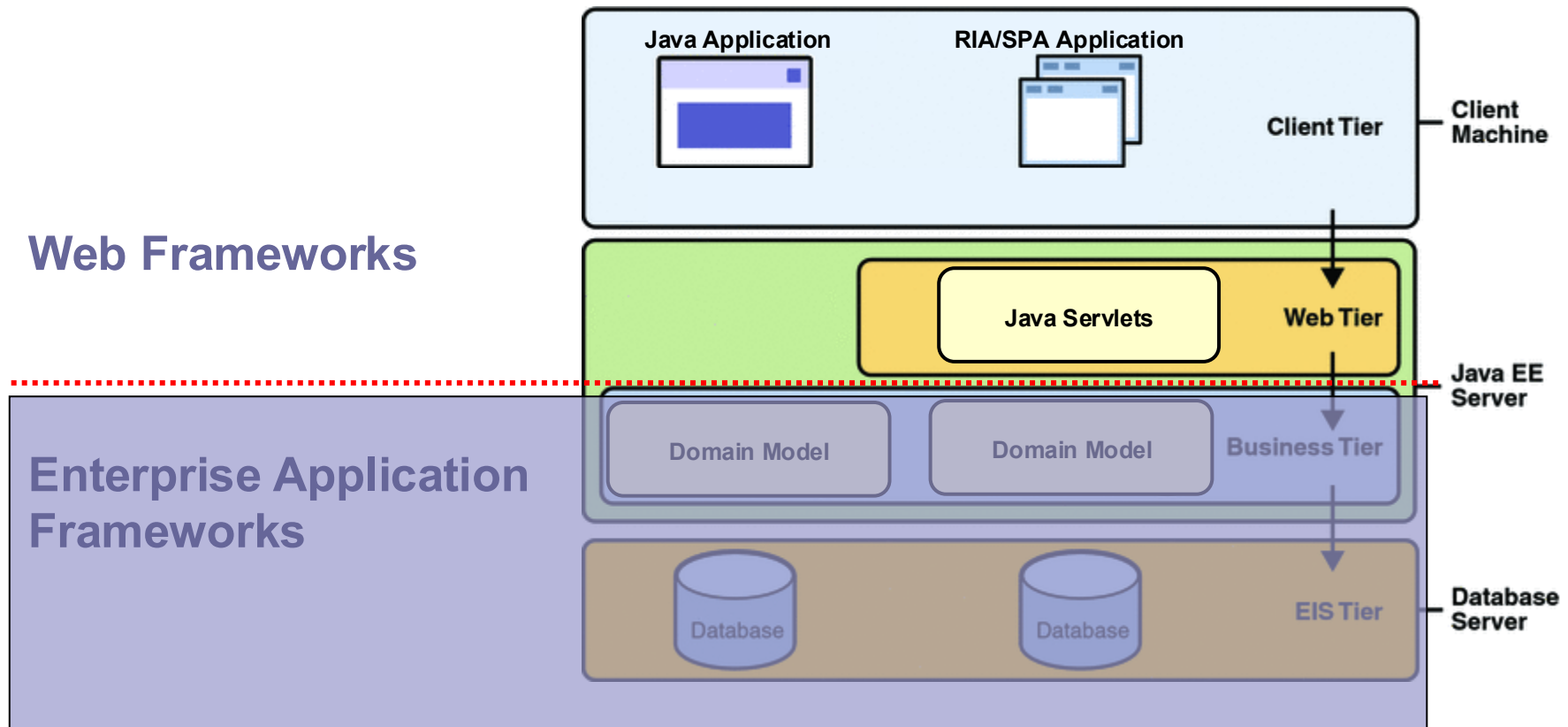


Image from Sun's Java EE Tutorial at <http://java.sun.com/javae/5/docs/tutorial/doc/bnaay.html>

# Wahl der Web Frameworks

- Serverseitige Technologien
  - SpringMVC
    - Request-Driven
  - Node/Express
    - Event-Driven
  - Thymeleaf
    - HTML Template Engine
    - Gute Integration mit SpringMVC
- Clientseitige Technologien
  - React
    - Single Page Application (SPA)
  - Zustand
    - Application State Management