

The background features a collage of various icons related to web development and technology. These include a hand cursor pointing at a screen, a magnifying glass over a globe, a target symbol, a document with a checkmark, a smartphone, a laptop, a gear, and a document with a large 'A'.

Web Entwicklung 2

MScIS Peter Grosskopf
MScIS Thomas Hollstegge

Fachbereich Wirtschaft - Fachhochschule Münster
Bachelor of Science Wirtschaftsinformatik Wintersemester 2013/2014

Agenda

- Organisatorisches
 - Veranstaltungsdaten
 - Vorstellung der Dozenten
 - Inhaltsübersicht
- Kapitel 0: Einleitung



Organisatorisches

Verlauf der Veranstaltung

- Vorlesungen und Übungen alle zwei Wochen abwechselnd
 - Zeit: Freitag 9h15 bis 12h30 + 13h15 bis 16h30
(= 4 Termine im Block)
 - Raum: Raum 1.10/11

Veranstaltungstermine

Woche	Datum	Veranstaltung
KW 42	18.10.2013	Vorlesung
KW 43	25.10.2013	-
KW 44	01.11.2013	-
KW 45	08.11.2013	Übung
KW 46	15.11.2013	Vorlesung
KW 47	22.11.2013	-
KW 48	29.11.2013	Übung
KW 49	06.12.2013	-
KW 50	13.12.2013	Vorlesung
KW 51	29.12.2013	-
KW 2	10.01.2014	Übung
KW 3	17.01.2014	-
KW 4	24.01.2014	Vorlesung + Übung

Leistungsnachweis / Klausur

- Leistungsnachweis: Fallstudie
- Ziel: Entwicklung einer Ruby-on-Rails-Anwendung
- Gruppenarbeit: bis 4 Personen
- Themen:
 - Auswahl eines vorgegebenen Themas oder eigener Vorschlag eines eigenen Themas
 - Vergabe erfolgt ca. Mitte des Semesters
- Abgabe:
 - Quellcodes
 - „Angemessene“ Dokumentation
- Abgabetermin:
 - Nach Ende der Klausurenphase (genauer Termin wird noch bekannt gegeben)

Kontext der Veranstaltung

Grundstufe		Aufbaustufe		P r a x i s p h a s e	Erweiterungsstufe	
1. Semester (Winter)	2. Semester (Sommer)	3. Semester (Winter)	4. Semester (Sommer)		5. Semester (Winter)	6. Semester (Sommer)
Grundlagen Wirtschaftsinformatik	Datenbanken	Betriebliche Anwendungssysteme I	Betriebliche Anwendungssysteme II		Projektmanagement	IT-Management
Grundlagen Programmierung	Software-Entwicklung I	Software-Entwicklung II	Software Engineering		Business Engineering	Projekt
Betriebssysteme und Rechnerarchitekturen	Web-Entwicklung I	Web-Entwicklung II	Web Engineering		E-Services	Thesis
Grundlagen Betriebswirtschaftslehre	Netzwerke	Betriebswirtschaftliche Primärprozesse	Wirtschaftsrecht		Business Intelligence	Kolloquium
Volkswirtschafts- lehre	Finanzwirtschaftl. Betriebswirtschaftslehre	Statistik	Quantitative Methoden		Schlüssel- kompetenzen III	
Wirtschafts- mathematik	Mathematik für Informatik	Englisch	Schlüssel- kompetenzen II		Wahlpflicht I	
Schlüssel- kompetenzen I			Transfermodul		Wahlpflicht II	

Lernziele

- Lernen, wie Programmierung im "Großen" geht
- Entwicklung eines grundlegenden Verständnis in Web- und Software Architekturen
- Lernen, wie man Projekte angeht und wie man Ideen online bekommt
 - Von der Idee ...
 - ... über Entwicklung ...
 - ... über Deployment ...
 - ... bis zur Maintenance

Programmierung im "Großen"

- Generell: Komplexe datenbankgestützte Webanwendungen mit Hilfe von serverseitigen Frameworks implementieren können
- Speziell:
 - Die Programmiersprache Ruby beherrschen
 - Das serverseitige Framework Ruby-on-Rails einsetzen können
 - Die Kenntnisse der clientseitigen Technologien HTML, CSS und JavaScript vertiefen
 - Aspekte der Usability bei der Entwicklung von Webanwendungen berücksichtigen können
 - Deployment von Webanwendungen
 - Aspekte der Sicherheit und Stabilität bei der Entwicklung von Webanwendungen berücksichtigen können

Didaktisches Konzept

- **Vorlesung**

- Vorstellung von Programmierkonzepten und Herangehensweisen
- Zunächst: Vermittlung von „theoretischem“ Wissen
- Später: Entwicklung eines komplexen Projektes (Step-by-Step)

- **Übung**

- Vertiefung von Inhalten der Vorlesung
- Praktische Anwendung des Wissens

- **Begleitmaterial**

- Orientierung an Railsguides (s.u.)
- Bei Bedarf Bereitstellung weiterer Materialien

Homepage der Veranstaltung

Auf Github

[https://github.com/Web-Entwicklung-2-WS-13-14/
Lectures](https://github.com/Web-Entwicklung-2-WS-13-14/Lectures)

oder

<http://zwtg.de/we2-13-14>

Literaturempfehlungen

- **Agile Web Development with Rails 4** by Sam Ruby, Dave Thomas, David Heinemeier Hansson <http://pragprog.com/book/rails4/agile-web-development-with-rails>
- **Programming Ruby 1.9 & 2.0 (4th edition): The Pragmatic Programmers' Guide** by Dave Thomas with Chad Fowler and Andy Hunt <http://pragprog.com/book/ruby4/programming-ruby-1-9-2-0>

Weblinks

- Ruby on Rails Website: <http://rubyonrails.org>
- Rails Guides: <http://guides.rubyonrails.org/>
- Gutes Videotutorial:
<http://railsforzombies.org>
- Screencasts zu verschiedenen Rails-Themen:
<http://railscasts.com>

Weitere Weblinks

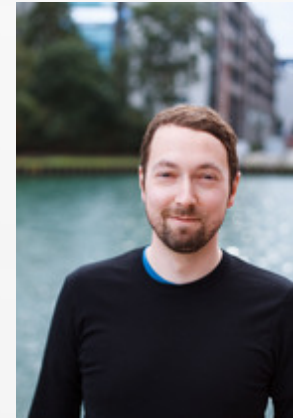
- <http://tryruby.org/levels/1/challenges/0>
- [http://mislav.uniqpath.com/poignant-guide/
book/chapter-1.html](http://mislav.uniqpath.com/poignant-guide/book/chapter-1.html)
- <http://api.rubyonrails.org/>
- <http://ruby-doc.org/core-2.0.0/>



Vorstellung der Dozenten

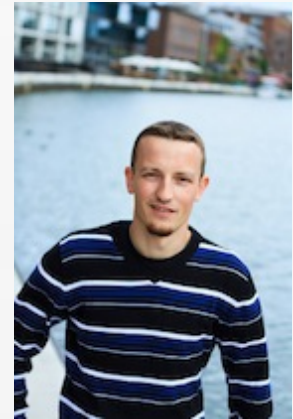
Peter Grosskopf (31)

- Geboren in Düsseldorf
- Studium der Wirtschaftsinformatik an der WWU Münster, Abschluss MScIS
- Während des Studiums:
 - Freelancing als Software Entwickler in Enterprise und Startup Firmen
 - Start eigener Projekte ...
 - ... z.B. Gründung von Zweitag
- https://www.xing.com/profile/Peter_Grosskopf



Thomas Hollstegge (27)

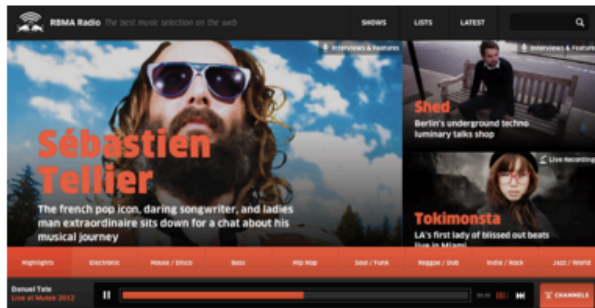
- Studium der Wirtschaftsinformatik an der WWU Münster, Abschluss MScIS
- Mitarbeiter bei Zweitag seit 2008
- Experte für Infrastruktur und Skalierung



Zweitag GmbH

- Beratung für Software Engineering aus Münster (<http://www.zweitag.de>)
- **Schwerpunkte**
 - Software-Entwicklung fürs Intra-/Internet
 - Konzeptentwicklung
 - Technologie-/Strategie-Beratung
- Geschäftsführer: Peter Grosskopf und Julian Schneider
- Gründung: Mai 2008
- Aktuell (10/13) 26 Mitarbeiter + Freelancer
- Technologien: Vor allem Ruby on Rails

Projekterfahrung (Auszug)

**RBMA RADIO**

Relaunch of Red Bull Music Academy Radio with [edenspiekermann](#)

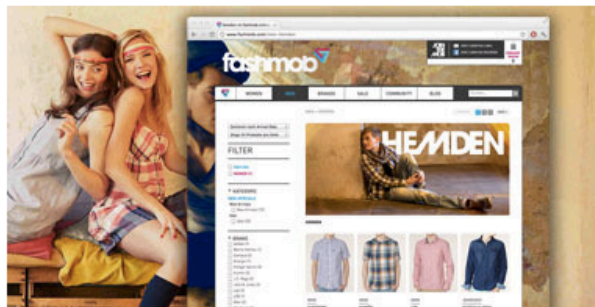


SO SCHMECKT'S

Custom shop solution built with Ruby on Rails

**SENTRES SOUTH TYROL**

Outdoor Tourism and Activity Portal for South Tyrol



FASHMOB.COM

Fashion community combined with shop and editorial section

**ZANOX**

Cloud deployment platform for the performance advertising company ZanoX

**REQORDER**

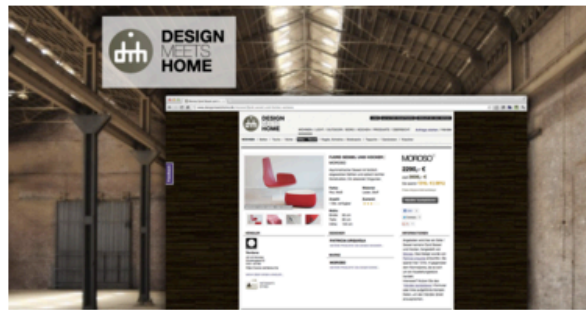
We provided additional resources to relaunch a major rework

Projekterfahrung (Auszug)



BLEACHERREPORT.COM

Partnering in Scrum-Team. B/R is based on Rails and other cutting edge technologies



DESIGN MEETS HOME

E-Commerce platform for design objects, based on Ruby on Rails and MySQL



SENTRES ALPINEMAP

Widget generator to create and publish activity maps. Based on node.js + couchDB



CELESIO TALENT REVIEW

Easy-to-use talent management tool based on (J)Ruby on Rails and Javascript



HANIEL

Intranet HR reporting application based on Ruby on Rails and SAP netweaver

Motivation (aus Dozentsicht)

- Web Entwicklung lernt man nicht aus Büchern, sondern in der Praxis
- Erfahrung und Wissen weiter geben
- Mehr Leute für moderne Technologien begeistern ("Evangelism")

Du? Du!? Du! Du!

- **Student 1:** Sie, Herr Grosskopf, können Sie ...
- **Peter Grosskopf:** <reagiert nicht> ;-)
- **Student 2:** Du, Peter, kannst du mir sagen, was in der letzten Vorlesung behandelt wurde?
- **Peter Grosskopf:** Klar, lieber Student 2 ... :-)



Inhaltsübersicht

Inhalte der Vorlesung

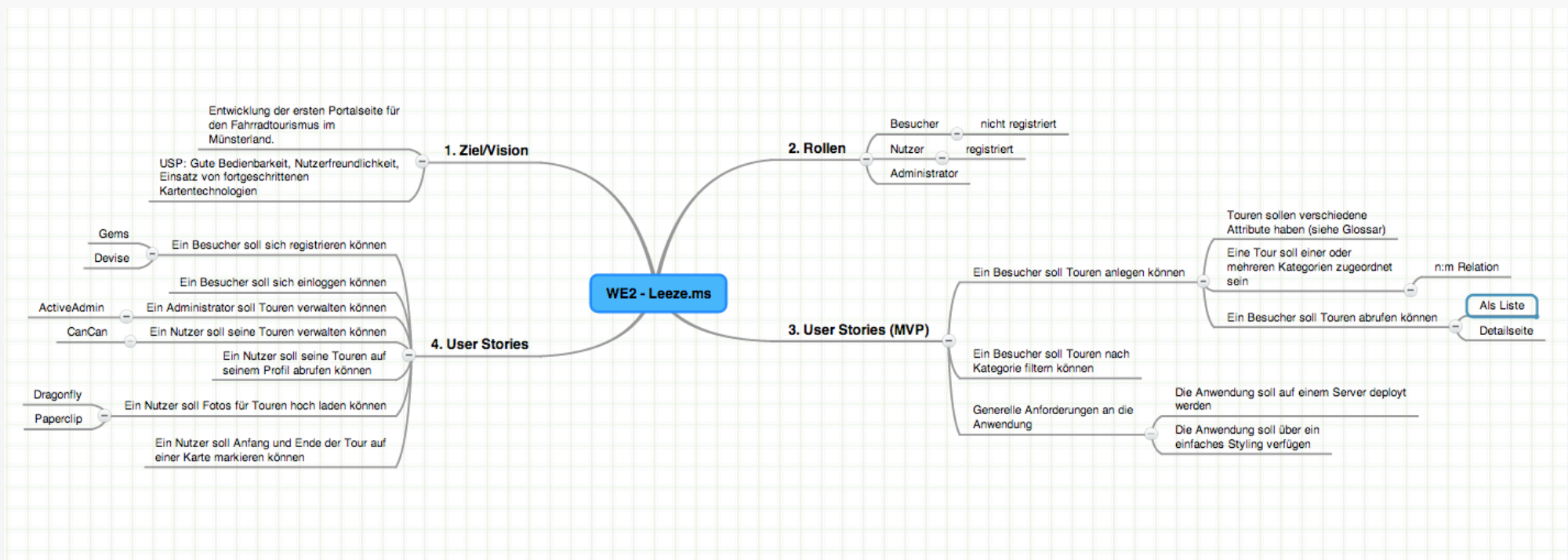
- Theorie-Teil

- Einleitung
- Einführung in Ruby on Rails

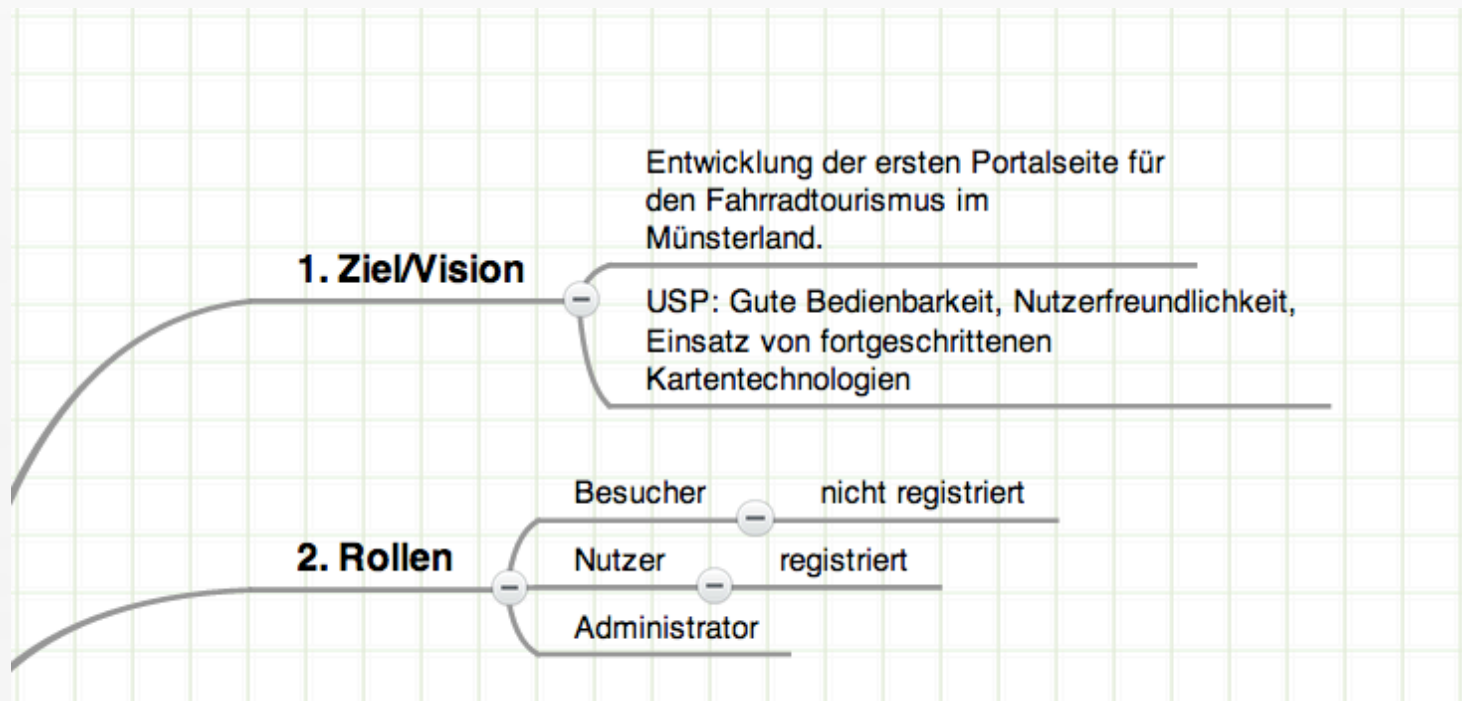
- Praxis-Teil

- Entwicklung der Fahrrad-Tourismus Plattform "leeze.ms"
- Agile Vorgehensweise mit Scrum

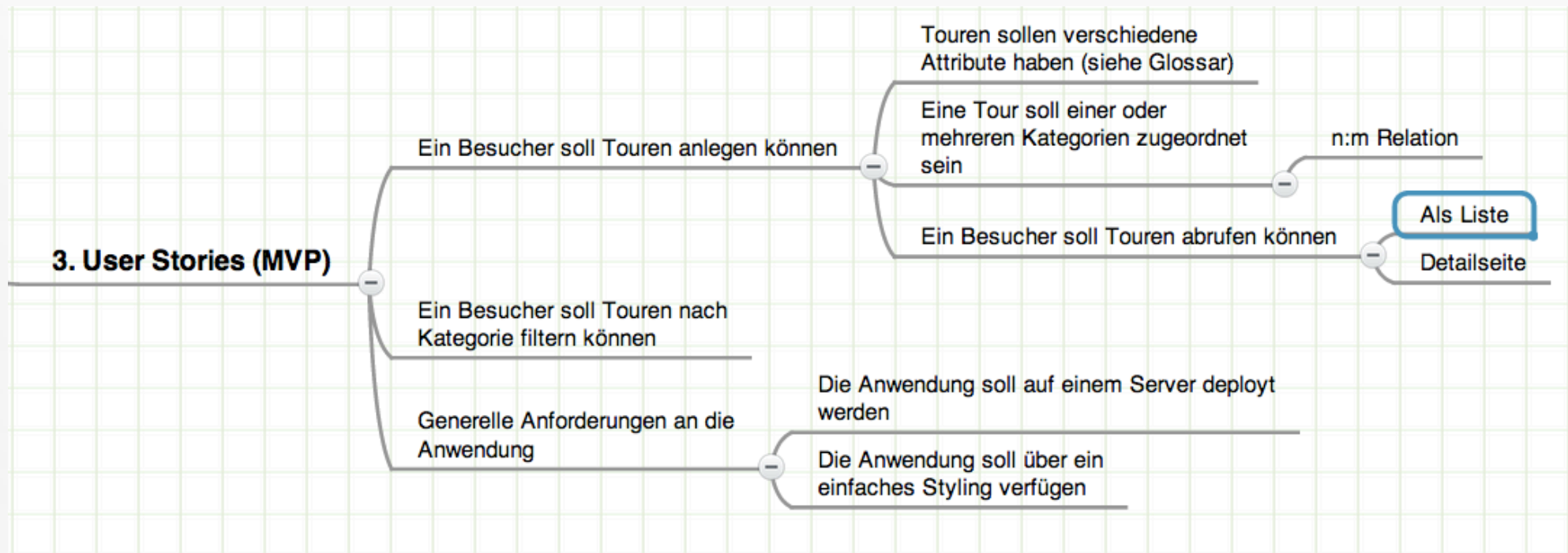
Leeze.ms



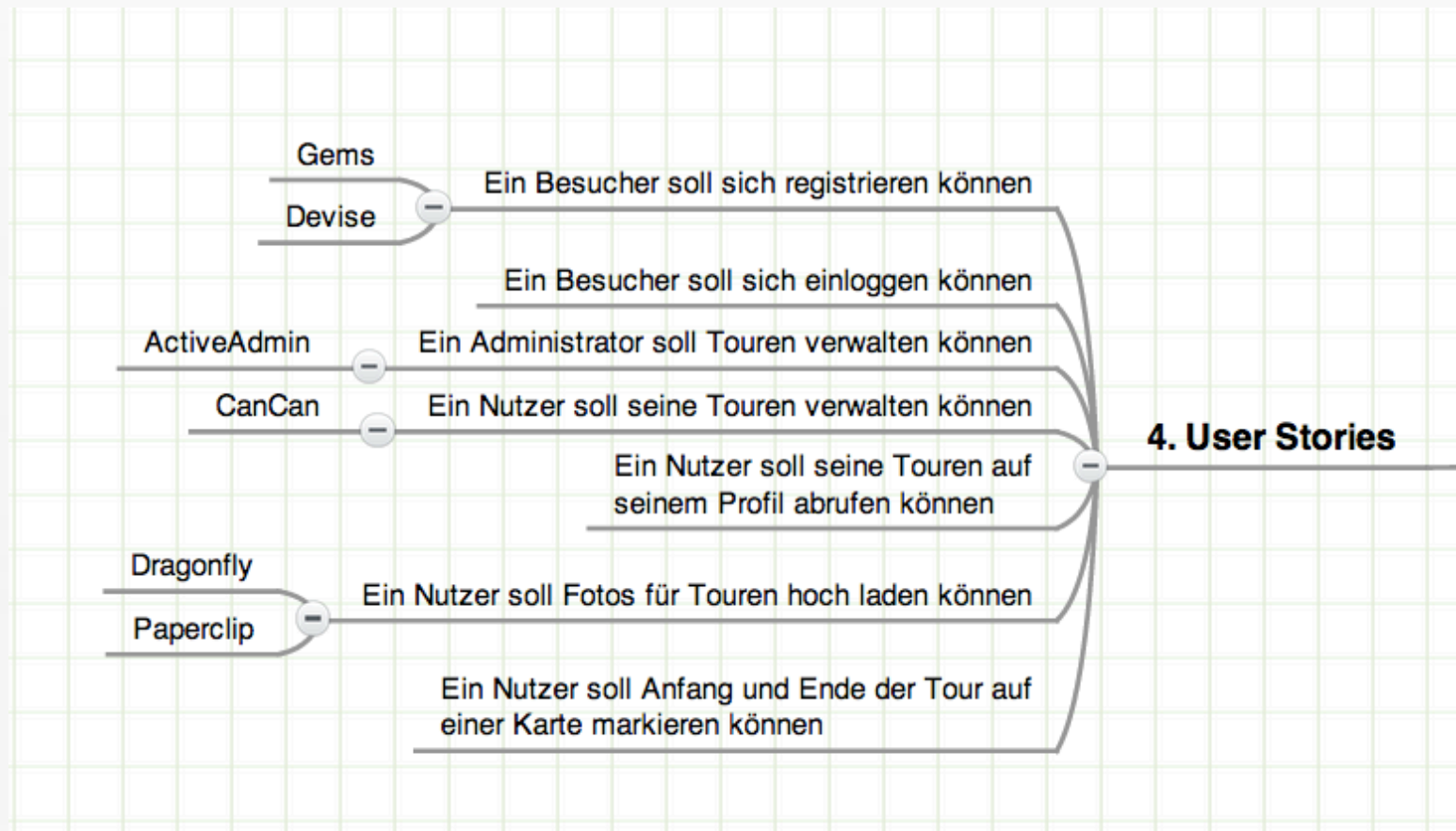
Leeze.ms



Leeze.ms



Leeze.ms





Kapitel 0: Einleitung

Ein kurzer Blick zurück ...

Fachhochschule
Münster University of
Applied Sciences



Webentwicklung I
Vorlesung

Prof. Dr.-Ing. Ingo Bax

Fachbereich Wirtschaft - Fachhochschule Münster

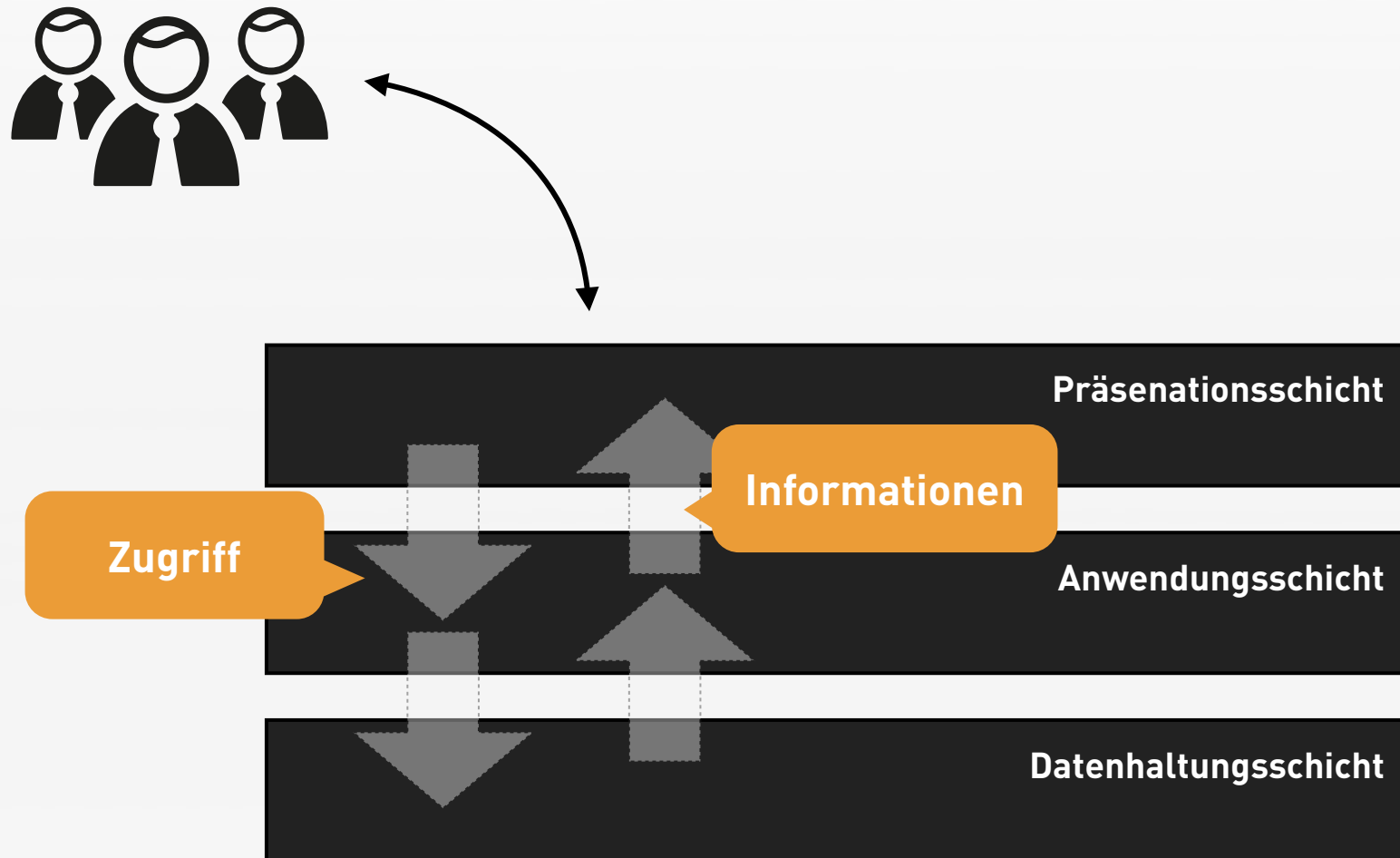
Bachelor of Science Wirtschaftsinformatik
Sommersemester 2012

Prof. Dr.-Ing. Ingo Bax Webentwicklung I - Vorlesung - SoSe 2012

Statische vs. dynamische Webseiten

- **Ursprünglicher Zweck des WWW:**
 - Darstellung von statischen Inhalten
 - Typisch: Server liefert den Inhalt einer auf der Festplatte abgelegten HTML-Datei aus
 - Üblich: Dokumente manuell erstellt
- **Technische Möglichkeit:**
 - Dynamische Erzeugung des auszuliefernden HTML-Dokumentes
 - Server-Software generiert Dokument „on-the-fly“, z.B. aus in einer Datenbank enthaltenen Informationen.
- **Konsequente Weiterentwicklung der Idee:**
 - Entkopplung der Präsentationsschicht einer Software-Anwendung über das WWW wird möglich

Schichten Modell



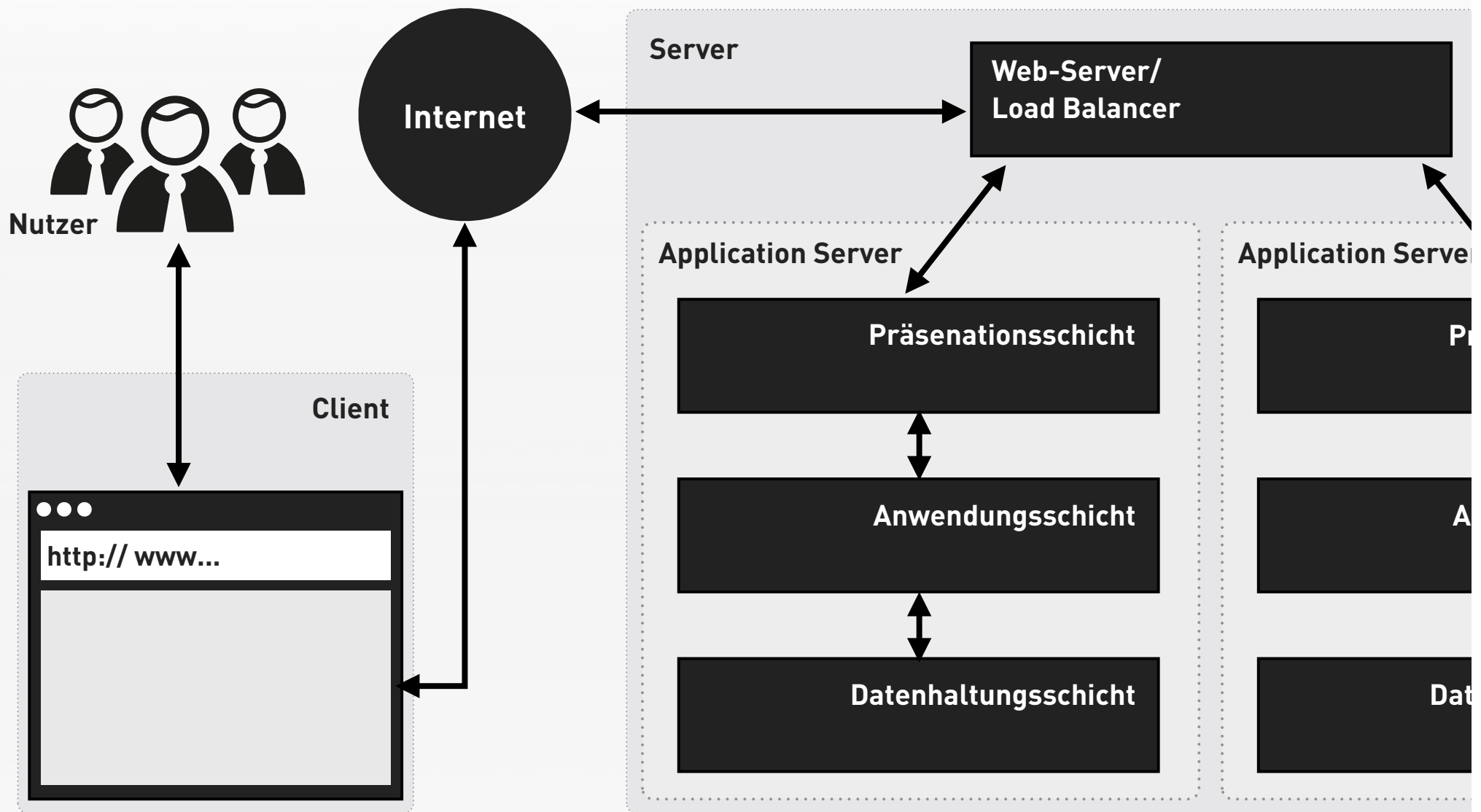
Schichten Modell

- **Präsentationsschicht (engl. Presentation Layer):**
 - Präsentation der Daten für den Benutzer
 - Entgegennahme von Benutzereingaben
 - Weitere Synonyme: Front-End, Client-Schicht, Benutzer-Schnittstelle
- **Anwendungsschicht (engl. Application Layer):**
 - Beinhaltet die eigentliche Geschäftslogik
 - Weitere Synonyme: Logik-Schicht, Middle-Tier, Enterprise-Tier, Businessschicht
- **Datenhaltungsschicht (engl. Data Layer):**
 - Verantwortlich für die persistente Speicherung der Daten
 - Weitere Synonyme: Back-End, Database-Tier

Client-Server

- Standard-Modell zur Verteilung von Aufgaben im Netzwerk
- Server:
 - Software-Anwendung, die **Dienst** (engl. Service) anbietet
 - Bereitstellung einer Netzwerk-Schnittstelle, über die der Dienst in Anspruch genommen werden kann
 - Festlegung eines **Protokolls**, welches den konkreten Ablauf und das Format des Datenaustausches definiert
 - **Passive Komponente**: Server wartet auf eingehende Anfrage (engl. **Request**), bearbeitet diese und sendet eine Antwort (engl. **Response**) zurück
- Client:
 - Software-Anwendung, die einen bestimmten Dienst nutzt
 - **Aktive Komponente**: Client stellt über das Netzwerk Anfrage an Server und wartet auf Antwort
 - Varianten: synchron vs. asynchron, Meldung vs. Auftrag

Architektur einer Web-Anwendung



HTTP Request Methods

- Werden von Representational State Transfer (REST) genutzt, um die Art der Anfrage zu beschreiben
- **GET**: fordert die angegebene Ressource vom Server an
- **POST**: legt eine neue Ressource auf dem Server an
- **PUT**: aktualisiert eine bestehende Ressource auf dem Server
- **DELETE**: löscht eine Ressource vom Server
- Weitere HTTP Request Methoden
 - PATCH, HEAD, OPTIONS

Web Technologien (Auswahl!)



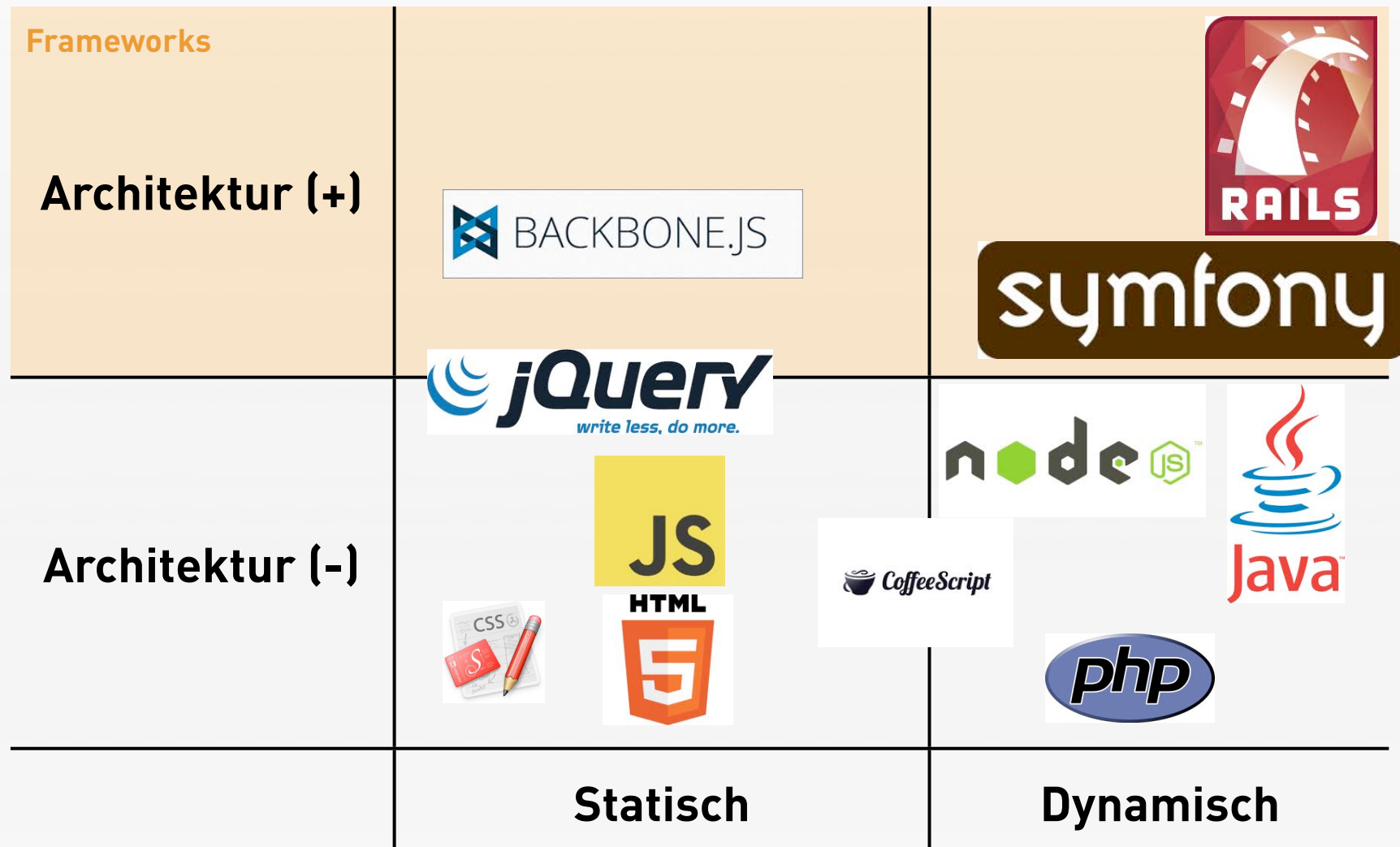
symfony



JS



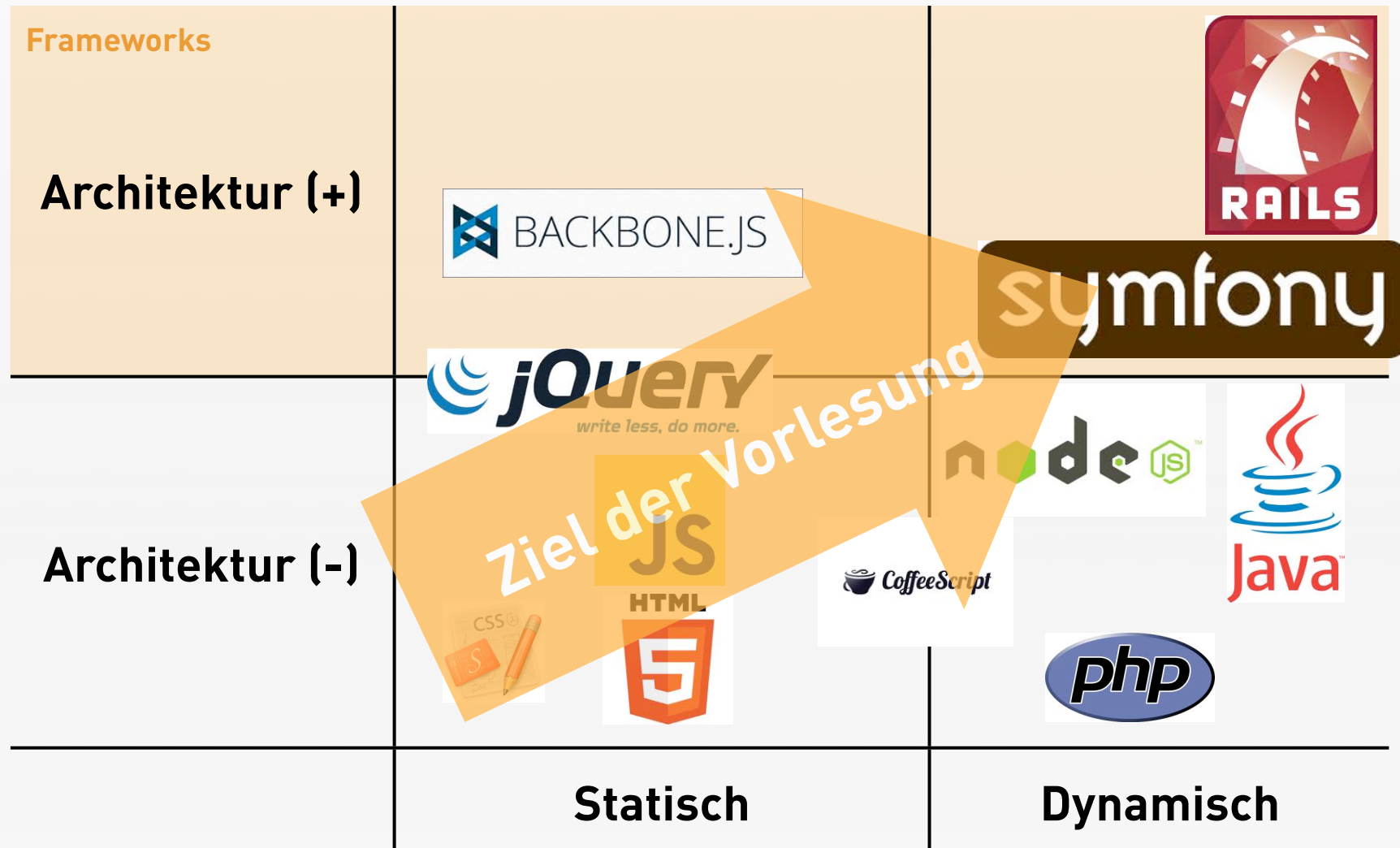
Web Technologien (Auswahl!)



Ziele von Frameworks

- Unterstützung bei Architekturentwurf
- Wiederholende Tätigkeiten vereinfachen
- Wiederverwendung von Code
- Standardisierung von Tätigkeiten („Dokumentation“)
- Steigerung der Entwicklungsgeschwindigkeit
- Nutzung von Best-Practises

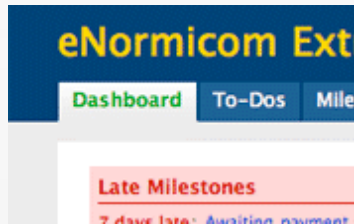
„Lernkurve“



Warum Ruby on Rails?

- Ruby: Moderne Programmiersprache
- Rails: Framework zur Webentwicklung
- Fokus ausschließlich auf Webanwendungen
- Vorteile Ruby on Rails
 - Große, aktive Community / Ecosystem
 - Viele Ressourcen im Netz
 - Best Practises
 - Steile Lernkurve
- Sehr viele, frei verfügbare Erweiterungen
 - z. B. für Anmeldung, Kommentare, etc.

Ruby on Rails Beispiele



Und viele mehr!

Fazit

- Los geht's! :-)