

W3WI_110 - Entwicklung verteilter Systeme

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum 149B
Version: 23SEB (SoSe 2025 - 1. Rev.)

Ausgewählte Inhalte gem. MHB - Web-Programmierung

Kerninhalte

- HTML, CSS, JavaScript als clientseitige Web-Technologien und aktuelle APIs (z.B. HTML5 und verwandte Technologien)
- Übertragungsprotokolle und APIs zwischen Client und Server (z.B. HTTP, HTTPS, WebSockets, XMLHttpRequest, Fetch API)
- Kommunikation zwischen einzelnen Komponenten Web-basierter Anwendungen
- Optimierung von Webseiten für verschiedene Zielsysteme

Zusatzinhalte

- Vertiefung von Frameworks
- Dynamische serverseitige Erzeugung von Webseiten

Ausgewählte Inhalte gem. MHB - Verteilte Systeme

- Terminologie, Konzepte, Architekturen, Anforderungsprofile und Architekturmodelle für verteilte Systeme
- Entwurfs- und Implementierungsansätze
- Vergleich unterschiedlicher Middleware-Konzepte
- Synchrone und asynchrone Kommunikation, entfernter Methodenaufruf
- Asynchrone Kommunikation und Messaging-Systeme
- Sicherheitsaspekte in verteilten Systemen

Prüfungsleistung - Portfolio

- das Modul hat 55 VL
- Verteilte Systeme hat 22VL, Web-Programmierung hat 33 VL

Zwei Bestandteile:

1. **(30 von 120 Punkten)** - Vorträge (Hinweise zur Präsentationen:
<https://delors.github.io/allg-vortraege/folien.de.rst.html>)
Die Präsentationen müssen am Abend vorher im Moodle hochgeladen werden!
2. **(90 von 120 Punkten)** - Projekt bzw. Programmieraufgabe in Teams von 4 Personen
 1. Projekt (Funktionsumfang, Code, Dokumentation, etc.)
 2. Abschlusspräsentation / Vorführung
 3. Code Reviews

Gruppenarbeit = Gruppennote



Das Projekt ist als Gruppenarbeit ausgelegt und alle Gruppenmitglieder erhalten für den Projektteil die gleiche Note.

Sollte eine individuelle Benoten gewünscht sein, dann ist dies **vor Abgabe des Projekts** explizit zu kommunizieren, damit das weitere Vorgehen besprochen werden kann.

Projekt/Programmieraufgabe

Entwicklung eines Kollaborationstools als wiederverwendbare Webkomponenten. D. h. jedes Team entwickelt *ein* eigenes Tool.

Beispiele:

- ein einfacher Chat
- ein einfaches interaktives Whiteboard
- ein Tool für ad-hoc Abstimmungen (zum Beispiel: wer hat die Hausaufgaben gemacht?)
- ein Tool für Umfragen
- ein Tool für Brainstorming/Mindmaps
- ein Tool für einfache Teameinteilungen
- ein Tool für die Erstellung von WordClouds
- ...

Idee ist immer, dass die Tools ohne explizite Anmeldung funktionieren. Es gibt keine Benutzerverwaltung. Die Anmeldung erfolgt über einen Link, der den Teilnehmern - zum Beispiel im Rahmen einer Vorlesung - zur Verfügung gestellt wird.

Es gibt eine *Adminoberfläche*, die es Admins - für diese Nutzer gibt es eine rudimentäre Verwaltung, die aber auch aus einer einfachen Konfigurationsdatei bestehen kann - ermöglicht neue Räume zu erzeugen und die Teilnehmer zu verwalten. Der Admin kann IP Adressen (temporär) sperren.

Bewertungskriterien für das Projekt

Kategorie	max. 90 Punkte
Code Reviews	max. 10
Abschlusspräsentation	max. 05
Vorführung	max. 10
Funktionsumfang	max. 15
Dokumentation (Entwickler und Benutzer)	max. 05
Qualität des Codes und der Tests (HTML, CSS und JavaScript)[1]	max. 40
Qualität des Buildprozesses[2]	max. 05

Es ist ein Dokument einzureichen aus dem hervorgeht:

1. welche KI Tools wofür eingesetzt wurden. (*Fehlanzeige erforderlich!*)
2. wer an welchem Teil mitgewirkt hat. (*Ohne dieses Dokument erfolgt keine Bewertung.*)

[1] Es ist neben dem Code auch ein kurzes Video 10 bis max. 15 Minuten einzureichen, dass in die Struktur und die Codebasis einführt. Dieses Video geht in die Benotung ein! Bitte nur im Notfall über Moodle bereitstellen.

[2] Werden Tests ausgeführt und wird am Ende ein Container gebaut?

Ablauf - W3WI-110 - Entwicklung verteilter Systeme 23SEB

- Scheduled: 16. May 2025 at 13:15 to 17:30, CEST
- Scheduled: 19. May 2025 at 13:15 to 17:30, CEST
- Scheduled: 22. May 2025 at 13:15 to 17:30, CEST
- Scheduled: 6. Jun 2025 at 13:15 to 17:30, CEST

Kurzpräsentation der Projekte (kein Code - Powerpoint ist ausreichend; 5 Minuten pro Team).
Kein unmittelbare Bewertung - dient "lediglich" zur Steuerung.

- Scheduled: 13. Jun 2025 at 13:15 to 17:30, CEST
- Scheduled: 16. Jun 2025 at 13:15 to 17:30, CEST
 - **Grundlagen der Virtualisierung** (Terminologie: z.B. Bare Metal Virtualisierung, Hypervisor Level.; Sicherheitsmodelle ggf. von CPU an.) - 2 Stud.
 - **Virtualization Platforms** (Proxmox und Openstack) - 2 Stud.
 - **Container Technologies** (Docker, Firecracker, Linux Containers (LXC)) - 2 Stud.
 - **Container Orchestrators** (Kubernetes, Docker Swarm) - 2 Stud.
- Scheduled: 24. Jun 2025 at 13:15 to 17:30, CEST
 - **Web- and Distributed Application Testing** (Diskussion und Präsentation von Werkzeugen für das Frontend- und Backend Testing) - 4 Stud.
(Jeder Studierende soll sich sein eigenes Thema suchen! D. h. es werden dann im Prinzip vier Einzelvorträge gehalten.)
 - **gRPC** und **gRPC-web** - 2 Stud.
 - **Apache Thrift** - 2 Stud.
- Scheduled: 30. Jun 2025 at 13:15 to 17:30, CEST
 - **Grundlagen von outdoor/indoor Positionierungssystemen** (GPS) - 1 Stud.
 - **Distributed Hash Tables** - 1 Stud.
 - **Paxos** - 2 Stud.
 - **Raft Consensus Algorithm** - 2 Stud.
 - **Gossip Protokoll** - 2 Stud.
- Scheduled: 4. Jul 2025 at 13:15 to 17:30, CEST

Gegenseitige Code Reviews und Präsentation der Ergebnisse der Code-Reviews.

(Jedes Team führt ein Review durch (45 Minuten) und wird auch einmal reviewed. Danach erstellen alle Teams einen Bericht über das Projekt, dass sie reviewed haben. Dafür stehen ca. 45 Minuten zur Verfügung. Die Berichte werden danach präsentiert (ca. 10 Minuten). Die Präsentation und die Berichte werden als Teil der Gruppenleistung bewertet.)
- Scheduled: 7. Jul 2025 at 13:15 to 17:30, CEST

Gruppenindividuelle Betreuung bei Fragen und Problemen bzgl. des Projekts.
- Scheduled: 11. Jul 2025 at 13:15 to 17:30, CEST

Abschlusspräsentationen (Vorstellung des Tools und Vorstellung wie die Komponenten genutzt werden kann - d. h. Code zeige) und Vorführung der Projekte.

Die Projektabgabe ist am 9.7.2025 um 23:59 Uhr.

Code Reviews - 4.7.2025

1. Durchführung eines Code Reviews (Frontend, Backend, Buildscripte, Projektstruktur, Dokumentation,...) eines anderen Projektes (2 * 45 Minuten)
Die Code Reviews erfolgen in zwei Runden, damit jede Gruppe ein Review bekommt; die Gruppen sollten sich aufteilen, damit alle Teile reviewt werden!
3. Erstellung eines Reports, der konstruktive Vorschläge enthält (30-45 Minuten).
4. Präsentation der Reports

-
- Gruppe 1: Planning Poker
Nico Wrede, Max Meinel, Okan Sönmez, Johannes Kling
 - Gruppe 2: CoCreate
Ramona Korten, Monika Pjano, Paulina Klaus, Lisa Molter
 - Gruppe 3: Chat App
Jonas Stammer, Felix Erhard, Luca Bäck, Raphael Plett
 - Gruppe 4: Chat App
Iven Stahl, Christian Zweigert, Ibrahim Tikce, Nils Teschke
 - Gruppe 5: Tool für Umfragen
Jonathan Wieder, Mika Jun, Leon Priemer, Sergio Meli
 - Gruppe 6: Planning Poker
Tom Weber, Tarnbir Singh, Jan Müller, Dilmand Sado