



Collaborative MarkDown Editor

Morten Terhart, Micha Spahr und Michael Skrzypietz
Duale Hochschule Baden-Württemberg Mosbach

11.06.2019



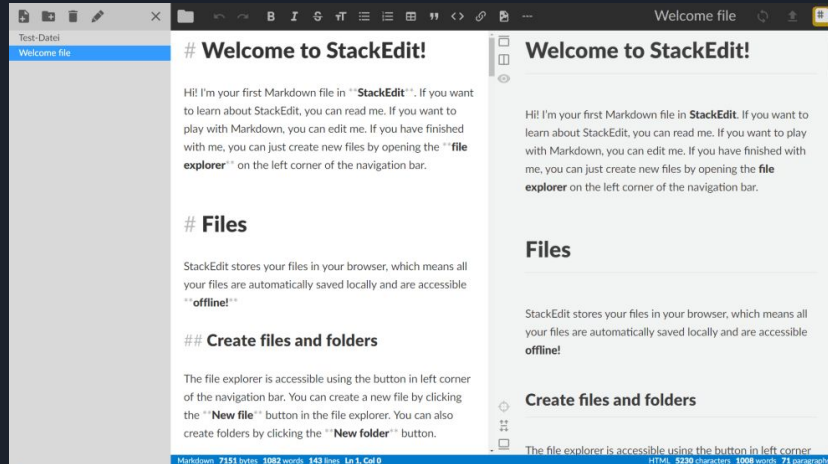
Agenda

1. Unsere Idee
2. Architektur und Technologien
3. Deployment mit Docker
4. Synchronisationsmodell
5. Demo

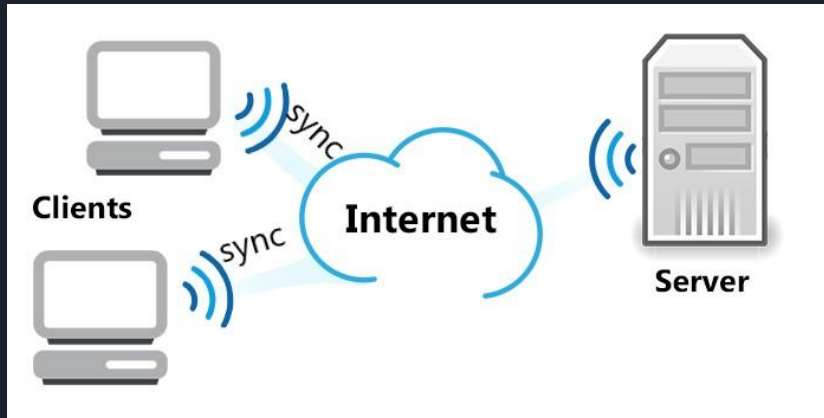
Unsere Idee



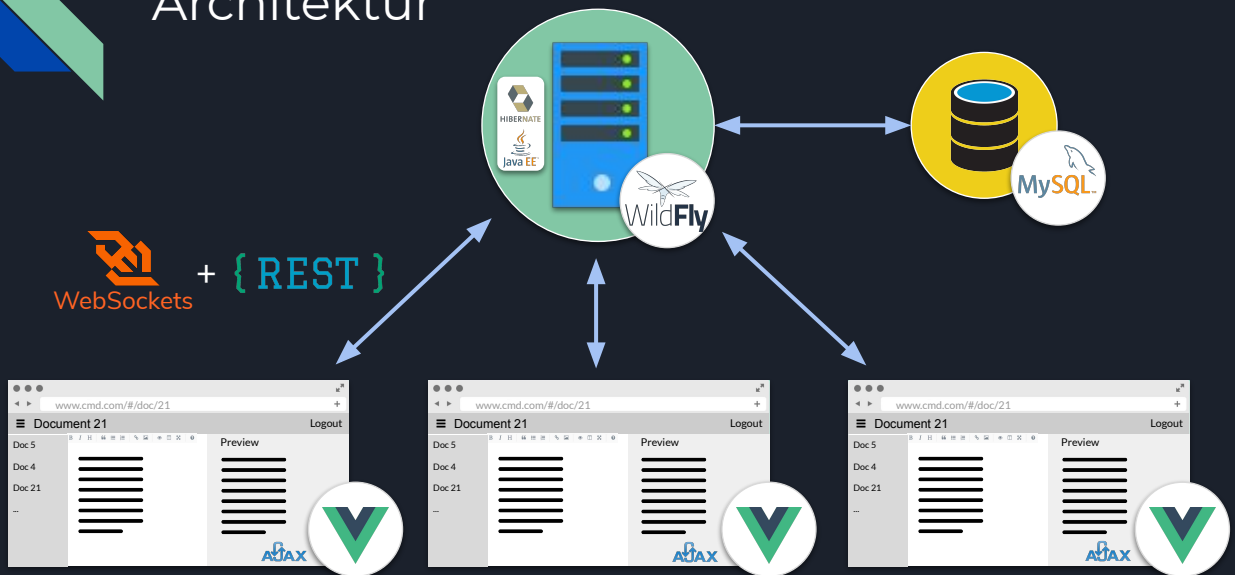
Unsere Idee



Unsere Idee



Architektur



Collaborative MarkDown Editor | Morten Terhart, Micha Spahr und Michael Skrzypietz | DHBW Mosbach | 11.06.2019

6

typische Client-Server-Architektur:

1. Webserver auf Basis von Wildfly mit Verbindung zur Datenbank, Webserver führt Java EE Stack aus
2. Client auf Basis des JavaScript-Frameworks Vue.js, jeder Client hat eigene Version des Dokuments, Kommunikation zwischen Server und Client über REST und WebSockets

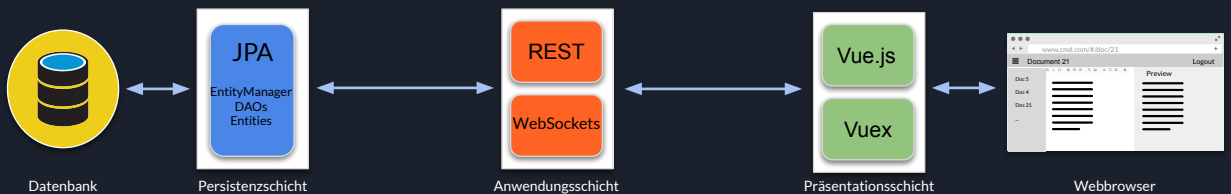
Technologien

Server: Java EE

- JPA: Hibernate + MySQL
- JAX-RS: RESTEasy + Jackson
- WebSocket Endpoint

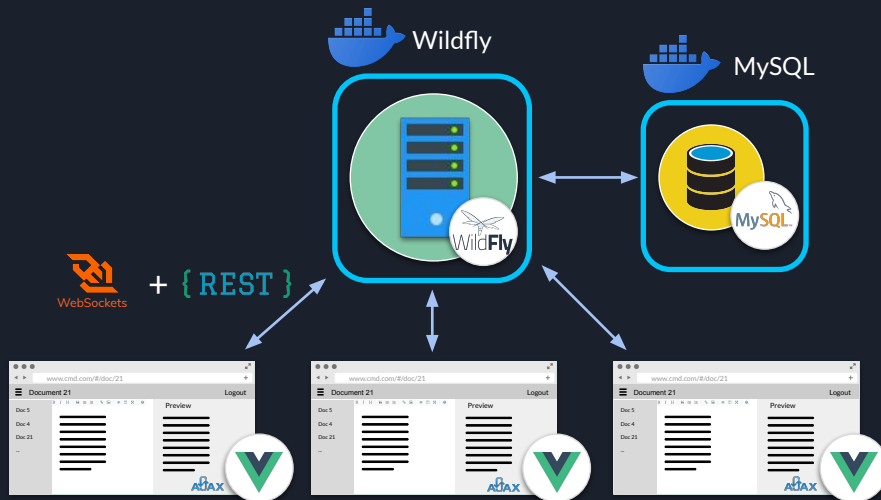
Client: Vue.js

- Vuex State Storage
- Axios (HTTP-Client)
- WebSocket Handler



- schichtenbasierte Anordnung der Software und Technologien
- **Persistenzschicht:** JPA
 - beinhaltet DAOs und Entities der Datenbank
 - stellt Verbindung zur Datenbank her
- **Anwendungsschicht:** REST und WebSockets
 - enthält REST-API für Dokumentenverwaltung
 - startet einen WebSocket-Endpoint für Synchronisation
- **Präsentationsschicht:** Vue.js
 - macht Aufrufe der REST-API und des WebSocket Endpoints
 - generiert die HTML-Repräsentation der Website

Deployment mit Docker



- Webserver und Datenbank in unterschiedlichen Containern
- können auf dem gleichen oder unterschiedlichen Rechnern ausgeführt werden
- kommunizieren über das Docker-Netzwerk miteinander

Synchronisationsmodell



Webbrowser



Austausch via
WebSockets



Server

- Transformationen
- Konsistenter Stand
- Fehlerbehebung
- Persistierung

Demo