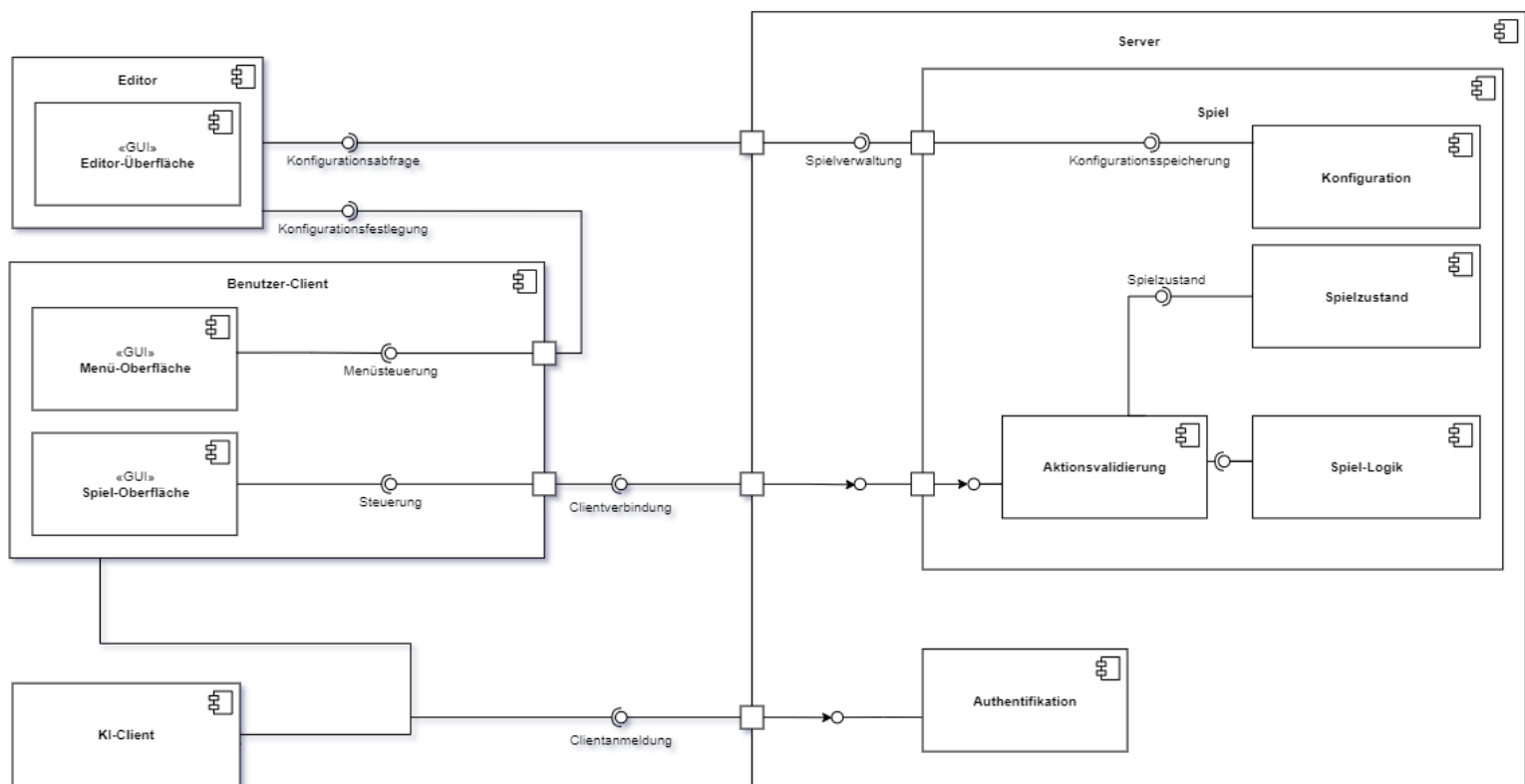


Meilenstein 5:

Komponentendiagramm:



Editor:

Die Komponente ist essentiell um individuelle Konfigurationen zu erstellen, die vom Spiel interpretiert werden. Diese können vom Server angefordert werden.

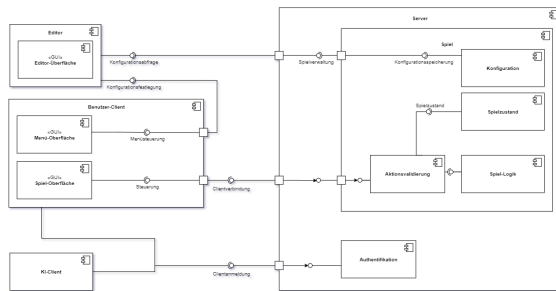
FAs: FA5

Editor-Oberfläche:

Die Oberfläche ist der Interaktionsbereich des Nutzers um den Editor zu steuern. Er hat eine eigene Komponente, da die Daten vom Nutzer eingegeben werden.

Die an der Oberfläche getätigten Eingaben werden vom Editor verarbeitet und bei einer Konfigurationsabfrage an den Server geschickt, wo die Dateien weiter interpretiert werden.

FAs: FA5



Benutzer-Client:

An dieser Komponente spielt sich die Eingabe des Spielers ab, heisst der Nutzer steuert was gemacht werden soll. Bei einer Front- & Backend Architektur handelt es sich hierbei um das Frontend.

FAs: FA1, FA3, FA11, FA34

Menü-Oberfläche:

Hier steuert der Nutzer über einzelne Menüpunkte. Er kann sich durch Menüs bewegen, den Editor starten und Menüpunkte auswählen.

FAs: FA25, FA35, FA36, FA37, FA38, FA40, FA44, FA45

Spiel-Oberfläche:

Der Spieler steuert hier verschiedene Aktionen des Spiels. Die Anweisungen werden verarbeitet und dann als Aktionen an den Server geschickt.

FAs: FA9, FA10, FA15, FA17, FA19, FA20, FA21, FA22, FA23, FA24, FA 26, FA27, FA28, FA29, FA32, FA33, FA39, FA43

KI-Client:

Client Anwendung die über einen Docker-Container läuft und Computergesteuert auch Anweisungen an den Server schickt.

FAs: FA4

Server:

Der Server hostet ein Spiel. Er authentifiziert dabei Clients. Außerdem kann er danach auch über Verbindungen Nachrichten entgegennehmen und an das Spiel weiterleiten. Hier werden die Verbindungen auch verwaltet.

FAs: FA1, FA2, FA8, FA16, FA18, FA30, FA31, FA32, FA41, FA42

Spiel:

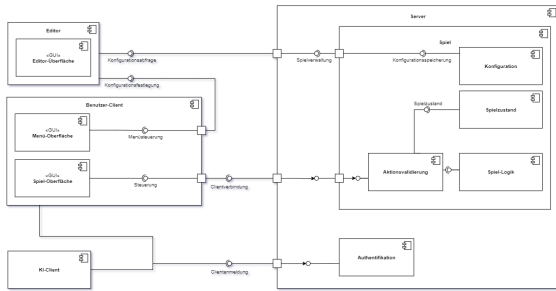
Das Spiel stellt die zentrale Spielsteuerung da. Hierbei werden alle Informationen verwaltet, die mit dem Spiel selber zu tun haben. Client Nachrichten und beliebige Konfigurationsdateien werden entgegengenommen und entsprechend verarbeitet.

FAs: FA14, FA15

Authentifikation:

Hier verwaltet der server anmeldungen der Spieler und KI clients. Angemeldete Spieler werden mit mit einer bestimmten ID versehen um sie eindeutig unterscheiden zu können.

FAs: FA37, FA41, FA42, FA45



Konfiguration:

Umfasst die 3 Jasons welche den Spielverlauf und die Helden definieren. Diese werden im Editor erstellt und bearbeitet. Sie werden vom Server geladen und können danach beliebig gelesen werden.

FAs: FA15

Spielzustand:

Der Spielzustand ist eine Speicherung des aktuellen Spiels und allen zugehörigen Werten. Er kann erst nach einer Authentifizierung der Nachricht verändert werden.

FAs: FA15

Aktionsvalidierung:

Jede einkommende Nachricht wird hier auf Korrektheit überprüft. Dadurch kann keine unmögliche, invalide Aktion ausgeführt werden. Wenn eine Nachricht als OK eingestuft wird, wird der Spielzustand dementsprechend verändert und eine Nachricht an alle Clients geschickt.

FAs: FA32

Spiel-Logik:

Die Logik umfasst alle im Spiel vorhandenen Vorgänge wie zum Beispiel den Helden bewegen oder Schaden zufügen. Jede von Clienten Aktion wird von der Spiellogik erfasst und ausgeführt.

FAs: FA32

Artefakte:

Charakter Konfiguration:

`characterConfig.json`

Partie Konfiguration:

`partie-konfig.json`

Szenario Konfiguration:

`szenario-name.scenario`, "`szenario-name`" muss ersetzt werden