



ulm university universität  
**uulm**

**Fakultät für  
Ingenieurwissenschaften,  
Informatik und  
Psychologie**

Institut für Software-  
technik und Program-  
miersprachen

# Softwaregrundprojekt Meilenstein 5

Softwaregrundprojekt an der Universität Ulm

**Vorgelegt von:**

Gruppe 10

**Dozent:**

Florian Ege

**Betreuer:**

Stefanos Mytilineos

2019

---

# Inhaltsverzeichnis

<b>I</b>	<b>Architekturentwurf</b>	<b>3</b>
<b>1</b>	<b>Leveleditor</b>	<b>3</b>
1.1	UML2-Komponentendiagramm . . . . .	3
1.2	Beschreibungen . . . . .	5
1.3	Zuordnung der Funktionalen Anforderungen . . . . .	5

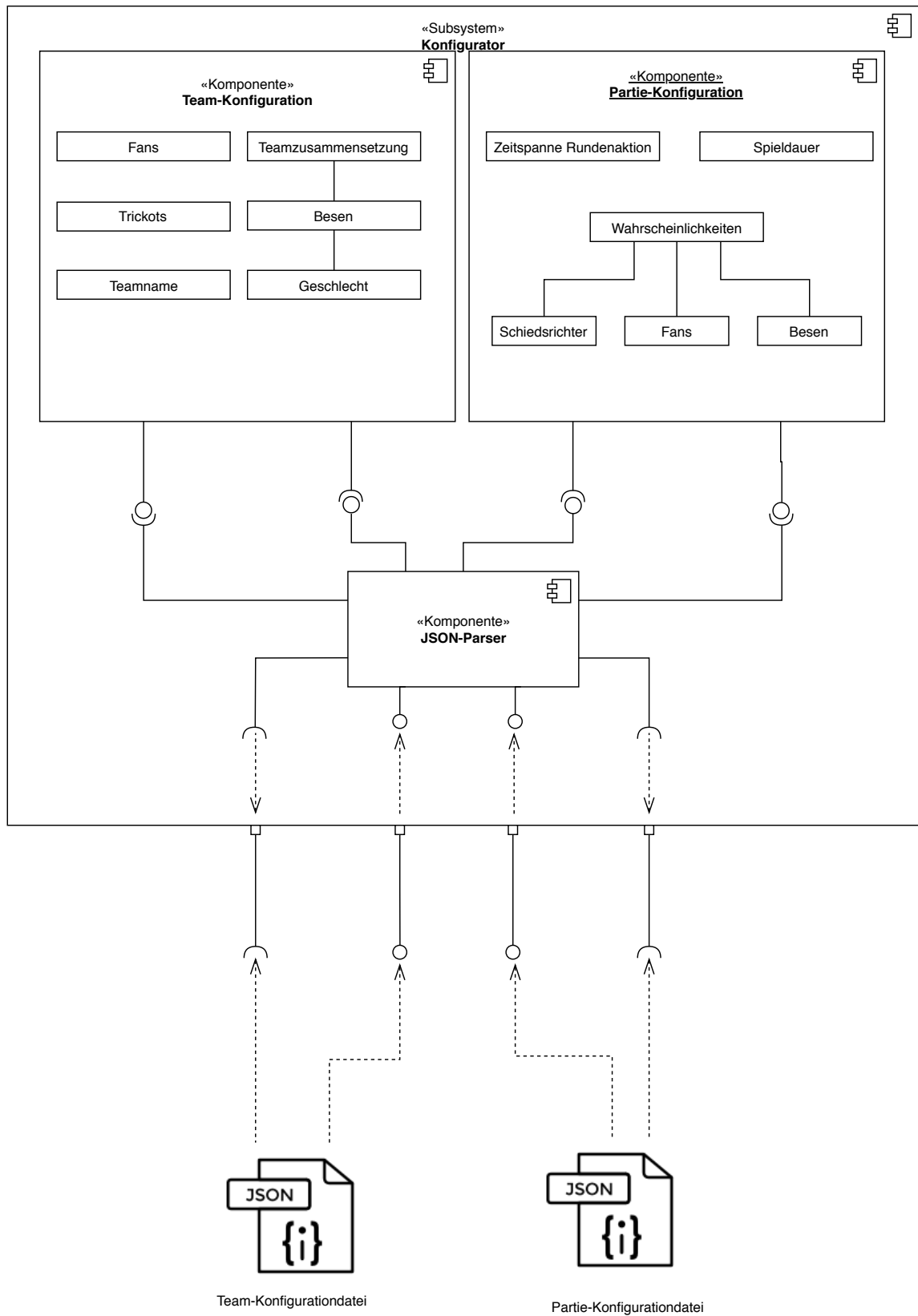
---

# **Teil I**

## **Architekturentwurf**

### **1 Leveleditor**

#### **1.1 UML2-Komponentendiagramm**



---

## 1.2 Beschreibungen

**Konfigurator** Bei dem Subsystem Konfigurator handelt es sich um eine graphische Benutzeroberfläche, welche zum erstellen von Partie-Konfigurationen, sowie zum erstellen von Team-Konfigurationen geeignet ist. Der Konfigurator ist lediglich zum erstellen einer gültigen Partie bzw Team-Konfiguration im JSON-Format da, ist aber ein eigenes Programm, und somit unabhängig von den anderen Anwendungen.

**Team-Konfiguration** Die Team-Konfiguration ist eine der Komponenten, die das Subsystem Konfigurator besitzt. In der Team-Konfiguration kann der Spieler sein Team erstellen, mit seinen gewünschten Parametern. Dies muss der Spieler nur einmal machen und anschliessend kann er seine Konfiguration abspeichern. Ebenso ist die Team-Konfiguration aber auch in der Lage, eine bereits vorhandene Konfiguration zu Laden und diese anschließend zu bearbeiten. Diese Komponente spielt somit eine entscheidende Rolle im Konfigurator, da über sie eine individuelle und gültige Team-Konfiguration erstellt werden kann.

**Partie-Konfiguration** Die Partie-Konfiguration ist eine Komponente des Subsystems Konfigurator und ist dazu gedacht, dass ein Spieler eine Partie mit seinen gewünschten Einstellungen erstellen kann. In der dafür vorgesehenen graphischen Oberfläche kann der Spieler dann seine Einstellungen leicht einstellen, welche von der Partie-Konfiguration in einem gültigen JSON-Format abgespeichert werden. Die Partie-Konfiguration kann ebenfalls eine bestehende Partie-Konfiguration laden, wo dass diese anschließend bearbeitet werden kann. Die Komponente ist spielt somit eine zentrale Rolle im Subsystem Konfigurator, da über sie erst eine gültige Partie-Konfiguration erstellt werden kann.

**JSON-Parser** Der JSON-Parser ist die letzte Komponente des Subsystems Konfigurator und dient dem abspeichern oder Laden der Team, bzw Partie-Konfiguration. Somit stellt der JSON-Parser eine eigene Komponente dar, über die die gewünschten Einstellungen für das Team oder die Partie abgespeichert werden, bzw geladen werden.

## 1.3 Zuordnung der Funktionalen Anforderungen

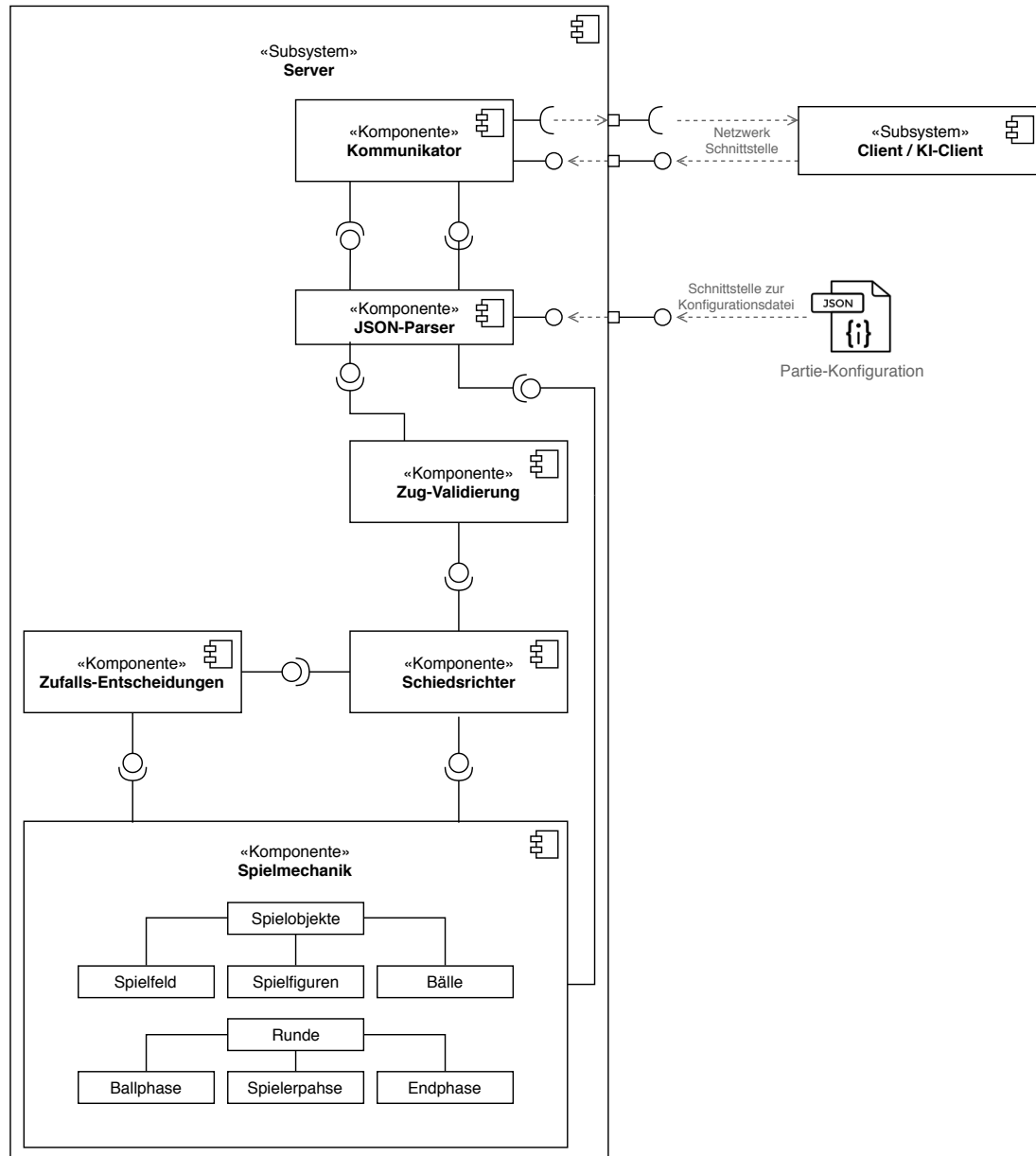
Die funktionalen Anforderungen gemäß dem Pflichtenheft werden den Komponenten folgendermaßen zugeteilt:

---

<b>Komponente</b>	<b>Abgedeckte funktionale Anforderungen</b>
Team-Konfiguration	FA53, FA70
Partie-Konfiguration	FA54, FA70
JSON-Konverter	FA71, FA72

## 2 Server

### 2.1 UML2-Komponentendiagramm



---

## 2.2 Beschreibungen

**Server** Beim Subsystem Server handelt es sich um die Konsolenanwendung, die als Schnittstelle für zwei oder mehr Clients dient. Die Serveranwendung kümmert sich dabei primär um die Spielsteuerung und agiert dabei unabhängig von den Client-Anwendungen.

**Spielmechanik** Die Spielmechanik ist die zentrale Komponente des Server Subsystems. Diese bildet dabei die komplette Partie intern ab. In der Spielmechanik sind also zu jeder Zeit die aktuellsten Status der Spielobjekte hinterlegt. Während eines Spiels senden die Clients ihre gewünschten Züge an den Server. Die Spielmechanik wertet die Züge, nachdem sie validiert und vom Schiedsrichter geprüft wurden, aus und aktualisiert dann gegebenenfalls die aktuelle Spielsituation. Diese wird im Anschluss wieder vom Server aus zu den einzelnen Clients ausgegeben. Als zentrale Komponente des Server Subsystems ist es zwingend notwendig, dass diese Komponente als Einheit gesehen wird, da ohne diesen Teil kein das Spielen nicht möglich wäre.

**Zug-Validierung** Die Zug-Validierung prüft, ob die Züge, die von einem Spieler über seinen Client übermittelt werden, grundsätzlich möglich sind. Dabei wird jedoch nicht geprüft ob der gewünschte Zug ein Foul darstellt. Notwendig ist diese Prüfung, da nicht sichergestellt ist, dass jede Client-Anwendung tatsächlich prüft ob die Züge die ein Spieler tätigen will auch grundsätzlich möglich sind. Die Komponente ist aus der eigentlichen Spielmechanik ausgegliedert, da die Zug-Validierung auch in anderen Subsystem eingesetzt werden könnte, z.b. einer Client-Anwendung.

**Schiedsrichter** Die Schiedsrichter Komponente des Server Subsystems. Möchte ein Spieler einen verbotenen Zug tätigen, wird in der Schiedsrichter Komponente die Entscheidung getroffen, ob der Spieler bestraft wird. Diese Komponente ist aus der Spielmechanik ausgegliedert, da da das Spiel grundsätzlich auch ohne diese Komponente möglich ist und daher diese Komponente in einem nachgelagerten Entwicklungsschritt implementiert werden kann.

**Zufalls-Entscheidungen** Dies Komponente stellt der Spielmechanik und der Schiedsrichter Komponente eine Schnittstelle zur Verfügung, die es erlaubt die vielen Zufalls-Ereignisse in einer Partie auszuwerten.

**Kommunikator** Der Kommunikator bildet die Komponente, die sich um die Datenübertragung zwischen der Server- und den Client-Anwendungen kümmert. Dabei werden dort zum einen die Status der Verbindungen zu den Clients überwacht und verwaltet. Zum anderen wird die Datenübertragung in beide Richtungen bereitgestellt. Da auch in den Client-Anwendungen eine ähnliche Komponente von Nöten ist, bietet es sich an, diese Funktionalitäten in einer eigenen Komponente auszulagern.

**JSON-Parser** Der JSON-Parser (de-) serialisiert die Objekte der Spiellogik. Da auch in den Client-Anwendungen eine ähnliche Komponente von Nöten ist, bietet es sich



---

an, diese Funktionalitäten in einer eigenen Komponente auszulagern.

## 2.3 Zuordnung der Funktionalen Anforderungen

Die funktionalen Anforderungen gemäß dem Pflichtenheft werden den Komponenten folgendermaßen zugeteilt:

<b>Komponente</b>	<b>Abgedeckte funktionale Anforderungen</b>
Spielmechanik	FA1 - FA36 FA43 - FA52 FA56 FA59
Schiedsrichter	FA36 - FA42
Zug-Validierung	FA67
Zufalls-Entscheidungen	FA58
Kommunikator	FA55
JSON-Konverter	FA53 FA57