



**Frankfurt University of Applied Sciences Fachbereich 2:
Informatik**

Bachelorthesis

zur Erlangung des akademischen Grades Bachelor of Science

**Einsatz von WebRTC für Rundenbasierte Brettspiele
im Vergleich zu Client-Server Architektur**

Autor: Robin Buhlmann

Matrikelnummer.: 1218574

Referent: Prof. Dr. Eicke Godehardt

Korreferent: Prof. Dr. Christian Baun

Version vom: 23. März 2021

Eidesstattliche Erklärung

Hiermit erkläre ich, Robin Buhlmann, dass ich die vorliegende Bachelorarbeit selbstständig und ohne unerlaubte Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Friedrichsdorf, den

Datum

Unterschrift

Abstract

Inhaltsverzeichnis

Eidesstattliche Erklärung	II
Abstract	III
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
Codeverzeichnis	VIII
Abkürzungsverzeichnis	IX
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	1
1.3 Aufbau der Arbeit	1
2 Technische Grundlagen	2
2.1 JavaScript	2
2.1.1 JavaScript-Bibliotheken	2
2.2 Node.js	2
2.3 WebRTC	2
2.4 CoTurn STUN / TURN Server	2
2.5 AWS	2
3 Design und Implementation der WebRTC–Infrastruktur	3
3.1 Analyse	3
3.2 Implementation der Peer-To-Peer Funktionalität	3
3.3 Implementation des Signaling-Servers	3
3.4 Aufsetzen und Konfiguration eines STUN und TURN Servers	3
4 Mensch–Ärgere–Dich–Nicht	4
4.1 Spielablauf	4
4.2 Analyse	4
4.3 Implementation	4
4.3.1 Darstellung des Spielbretts	4
4.3.2 Implementation der Spiellogik	4

4.3.3	Synchronisation des Spielstands	4
4.3.4	Probleme der Implementation	4
5	Evaluation	5
6	Zusammenfassung und Ausblick	6
7	Literaturverzeichnis	7

Abbildungsverzeichnis

Tabellenverzeichnis

Codeverzeichnis

Abkürzungsverzeichnis

1 Einleitung

1.1 Motivation

1.2 Zielsetzung

1.3 Aufbau der Arbeit

2 Technische Grundlagen

2.1 JavaScript

2.1.1 JavaScript-Bibliotheken

socket.io

crypto.js

2.2 Node.js

2.3 WebRTC

2.4 CoTurn STUN / TURN Server

2.5 AWS

3 Design und Implementation der WebRTC–Infrastruktur

3.1 Analyse

3.2 Implementation der Peer-To-Peer Funktionalität

3.3 Implementation des Signaling-Servers

3.4 Aufsetzen und Konfiguration eines STUN und TURN Servers

4 Mensch-Ärgere-Dich-Nicht

4.1 Spielablauf

4.2 Analyse

4.3 Implementation

4.3.1 Darstellung des Spielbretts

4.3.2 Implementation der Spiellogik

4.3.3 Synchronisation des Spielstands

4.3.4 Probleme der Implementation

Look-Ahead

Faire Zufallszahlen

5 Evaluation

6 Zusammenfassung und Ausblick

7 Literaturverzeichnis