





Hochschule für Technik,  
Wirtschaft und Kultur Leipzig

**Fakultät Informatik und Medien**

Studiengang Medieninformatik

# **Vergleichende Analyse der komponentenbasierten Frontend-Frameworks Angular und React**

Bachelorarbeit

zur Erlangung des akademischen Grades

**Bachelor of Science**

vorgelegt von

Felix Schmeißer

geb. am 22.02.1999

in München

69578

Verantwortlicher Hochschullehrer: Prof. Dr. rer. nat. Klaus Hering  
Leipzig, Juni 2020 – September 2020

## Erklärung

Ich versichere wahrheitsgemäß, diese Arbeit selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

.....

Felix Schmeißer

Leipzig, den 9. Juli 2020

## Danksagung

Zunächst möchte ich meinem Betreuer Prof. Dr. rer. nat. Klaus Hering danken.

:

Zuletzt danke ich meinen Freunden, meinen Eltern sowie meiner Familie für die ständige Unterstützung während meines Studiums.

# Inhaltsverzeichnis

<b>Abbildungs- und Tabellenverzeichnis</b>	<b>6</b>
<b>1 Einleitung</b>	<b>7</b>
1.1 Problemstellung	7
1.2 Ziel der Arbeit	7
1.3 Vorgehensweise	8
1.4 Begriffsabgrenzung	8
1.5 Inhaltlicher Aufbau der Arbeit	8
<b>2 Allgemeines</b>	<b>9</b>
2.1 Entstehung	10
2.2 Verbreitung und Beliebtheit	10
2.3 Verwendete Sprachen	10
2.3.1 JavaScript	10
2.3.2 HTML-Templates	10
2.3.3 CSS	10
2.4 Projekterstellung	10
2.4.1 Vorgehensweise	10
2.4.2 Aufbau	10
<b>3 Technischer Vergleich</b>	<b>11</b>
3.1 Components	12
3.1.1 Komponentenmodell	12
3.1.2 Angular	12
3.1.3 React	12
3.2 Weitere Features	12
3.2.1 Angular	12
3.2.2 React	12
3.3 Lifecycle	12
3.3.1 Angular	12
3.3.2 React	12
3.4 DOM	12
3.5 Integrierte Design Pattern	12
3.5.1 Dependency Injection	12

---

3.5.2	Services . . . . .	12
3.5.3	Decorators . . . . .	12
3.5.4	Pipes . . . . .	12
3.5.5	State Management . . . . .	12
3.6	Projektarchitekturen . . . . .	12
<b>4</b>	<b>Implementation . . . . .</b>	<b>13</b>
4.1	Anforderungsbeschreibung . . . . .	13
4.1.1	Aufbau . . . . .	13
4.1.2	Identische Teile . . . . .	13
4.1.3	Umsetzungsdetails . . . . .	13
4.2	Angular . . . . .	13
4.3	React . . . . .	13
<b>5</b>	<b>Performance Test . . . . .</b>	<b>14</b>
5.1	Testszenarien . . . . .	14
5.2	Durchführung . . . . .	14
<b>6</b>	<b>Auswertung . . . . .</b>	<b>15</b>
6.1	Grundsätzlicher Vergleich . . . . .	15
6.2	Probleme (Implementation) . . . . .	15
6.3	Auswertung der Performance-Tests . . . . .	15
6.4	Handlungsempfehlung . . . . .	15
6.4.1	Anwendungsbereiche . . . . .	15
6.4.2	Lernkurve . . . . .	15
<b>7</b>	<b>Fazit . . . . .</b>	<b>16</b>
7.1	Erfahrungen . . . . .	16
7.2	Lernerfolge . . . . .	16
7.3	Ausblick . . . . .	16
<b>Anhang</b>		<b>18</b>

---

# Abbildungs- und Tabellenverzeichnis

## Abbildungsverzeichnis

Abbildung 2.1: Bildunterschrift . . . . .	9
---	---

## Tabellenverzeichnis

Tabelle 2.1: Tabellenüberschrift . . . . .	9
--	---

# 1 Einleitung

## 1.1 Problemstellung

Es gibt unzählige JavaScript-Frameworks und es kommen ständig neue hinzu. Als Entwickler muss man abwägen, welche Lösung für das speziell vorliegende Szenario geeignet ist.

Angular ist ein umfangreiches Frontend-Framework und kann damit nahezu jede Aufgabe in diesem Bereich abdecken. Für sehr viele Problemstellungen gibt es eine Lösung direkt aus dem Framework. Die vorgesehene Architektur ist MVC bzw. MVVM und forciert damit eine strikte Trennung, die Entwicklung in großen Teams vereinfacht.

React als JavaScript-Bibliothek ist deutlich reduzierter. Abseits der elementaren Funktion von React werden Community-Erweiterungen verwendet. Demzufolge muss man hier zwischen verschiedenen Lösungsmöglichkeiten abwägen. React verwendet mit JSX eine JavaScript-Erweiterung, welche HTML und JavaScript kombiniert. Das macht die Entwicklung von Komponenten deutlich schneller, bricht allerdings mit MV\*-Architekturen.

## 1.2 Ziel der Arbeit

Das Ziel der Arbeit ist, Gemeinsamkeiten und Unterschiede zwischen den Technologien herauszuarbeiten, auch mögliche Vor- und Nachteile hinsichtlich der Architektur werden angeschnitten. Dazu wird im Rahmen der Arbeit eine Testanwendung mit den Frameworks

---



implementiert. Durch Steuerung der Datenmenge und künstliche Geschwindigkeitsdrosselung können verschiedene Szenarien simuliert werden, um die Anwendungsbereiche der Frameworks einzugrenzen.

Welche Gemeinsamkeiten und Unterschiede besitzen Angular und React und welche Anwendungsbereiche ergeben sich aus der Performance in unterschiedlichen Auslastungsszenarien?

## **1.3 Vorgehensweise**

Zunächst ein theoretischer Vergleich beider Technologien, der die grundlegenden Features beschreibt und gegenüberstellt. Im zweiten Schritt wird der Vergleich praktisch durchgeführt. Ziel ist es, eine identische Anwendung einmal in Angular und React zu implementieren, um die Unterschiede zu verdeutlichen und Grenzen aufzuzeigen. Im Anschluss werden verschiedene Testszenarien ausgewertet, um Anforderungen kleiner Anwendungen und größerer Enterprise-Produkte zu vergleichen. Abschließend werden die Beobachtungen eingeordnet und Schlüsse über Vor- und Nachteile beider Technologien gezogen. Zudem werden kurz Lösungsmöglichkeiten für etwaige Probleme diskutiert, damit einher geht ein Ausblick über die Erweiterbarkeit und Einbindung von externen Lösungen.

## **1.4 Begriffsabgrenzung**

Diese Arbeit geht auf die essentiellen Funktionalitäten beider Frameworks ein und stellt die entsprechende Umsetzung im jeweils anderen vor. Fortgeschrittene Themen und Erweiterungen werden nicht genauer betrachtet, werden aber mit Verweis auf offizielle Quellen in die Argumentation eingebunden.

## **1.5 Inhaltlicher Aufbau der Arbeit**

---

## 2 Allgemeines

Ein Textbeispiel. [1]

Ein neuer Absatz und ein weiteres Zitat [2].

Abbildung 2.1: Bildunterschrift

Tabelle 2.1: Tabellenüberschrift

## **2.1 Entstehung**

## **2.2 Verbreitung und Beliebtheit**

## **2.3 Verwendete Sprachen**

### **2.3.1 JavaScript**

TypeScript

JavaScript mit Flow

### **2.3.2 HTML-Templates**

HTML-Templates (Angular)

JSX (React)

### **2.3.3 CSS**

## **2.4 Projekterstellung**

### **2.4.1 Vorgehensweise**

### **2.4.2 Aufbau**

---



## 3 Technischer Vergleich

### 3.1 Components

#### 3.1.1 Komponentenmodell

#### 3.1.2 Angular

#### 3.1.3 React

### 3.2 Weitere Features

#### 3.2.1 Angular

#### 3.2.2 React

### 3.3 Lifecycle

#### 3.3.1 Angular

#### 3.3.2 React

### 3.4 DOM

Regular DOM

Virtual DOM

Vergleich

### 3.5 Integrierte Design Pattern

---

#### 3.5.1 Dependency Injection

#### 3.5.2 Services

#### 3.5.3 Dependency

## **4 Implementation**

### **4.1 Anforderungsbeschreibung**

#### **4.1.1 Aufbau**

#### **4.1.2 Identische Teile**

#### **4.1.3 Umsetzungsdetails**

**Verwendete Features**

**Backend-Mockup**

### **4.2 Angular**

### **4.3 React**

---

## **5 Performance Test**

### **5.1 Testszenarien**

### **5.2 Durchführung**

---

## **6 Auswertung**

### **6.1 Grundsätzlicher Vergleich**

### **6.2 Probleme (Implementation)**

### **6.3 Auswertung der Performance-Tests**

### **6.4 Handlungsempfehlung**

#### **6.4.1 Anwendungsbereiche**

#### **6.4.2 Lernkurve**

---



## **7 Fazit**

### **7.1 Erfahrungen**

### **7.2 Lernerfolge**

### **7.3 Ausblick**

---

# Literatur

- [1] Joachim Schlosser. *Wissenschaftliche Arbeiten schreiben mit L<sup>A</sup>T<sub>E</sub>X*. 5. Aufl. MITP-Verlags GmbH & Co. KG, 2013. ISBN: 9783826659355.
- [2] Moritz Nadler. *ISO-31-konformer Formelsatz in L<sup>A</sup>T<sub>E</sub>X Version 1.0*. 17. Dez. 2015. URL: <http://www.moritz-nadler.de/formelsatz.pdf>.

# Anhang

# Anhangsverzeichnis

<b>Abbildungs- und Tabellenverzeichnis im Anhang . . . . .</b>	<b>20</b>
<b>A Anhangskapitel . . . . .</b>	<b>21</b>

# Abbildungs- und Tabellenverzeichnis im Anhang

## Abbildungen im Anhang

Abbildung A.1: Bildunterschrift im Anhang . . . . .	21
---	----

## Tabellen im Anhang

Tabelle A.1: Tabellenüberschrift im Anhang . . . . .	21
--	----

# A Anhangskapitel

Abbildung A.1: Bildunterschrift im Anhang

Tabelle A.1: Tabellenüberschrift im Anhang

