

**Proposal** zur Bachelorarbeit

# Bessere Nachvollziehbarkeit von JavaScript-Anwendungen im Browser

**Ansatz zur Verbesserung der Nachvoll-  
ziehbarkeit von Nutzerinteraktion und  
Anwendungsverhalten**

An der Fachhochschule Dortmund  
im Fachbereich Informatik  
Studiengang Software- und Systemtechnik, Vertiefung Softwaretechnik  
erstelltes **Proposal** zur Bachelorarbeit  
zur Erlangung des akademischen Grades  
Bachelor of Science

von  
Marvin Kienitz  
geb. am 26.04.1996  
Matr.-Nr. 7097533

Betreuer:  
Prof. Dr. Sven Jörges  
Dipl. Inf. Stephan Müller

Dortmund, 4. Dezember 2020

## Kurzfassung

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

*Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.*

*Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.*

*Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.*

## Abstract

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

*Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.*

*Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.*

*Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.*

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung . . . . .	2
1.2.1	Abgrenzung . . . . .	2
1.3	Vorgehensweise . . . . .	3
1.4	Open Knowledge GmbH . . . . .	3
<b>2</b>	<b>Ausgangssituation</b>	<b>4</b>
2.1	Browserumgebung . . . . .	4
2.1.1	Browserprodukte . . . . .	4
2.1.2	JavaScript . . . . .	4
2.2	Clientbasierte Webapplikationen . . . . .	4
2.2.1	JavaScript-basierte Webapplikationen . . . . .	4
2.2.2	Single-Page-Applications . . . . .	4
2.3	Nachvollziehbarkeit . . . . .	5
2.3.1	Nutzen . . . . .	5
2.3.2	Nachvollziehbarkeit bei SPAs . . . . .	5
2.3.3	Browserbedingte Hürden . . . . .	5
<b>3</b>	<b>Methoden und Praktiken</b>	<b>6</b>
3.1	Methoden . . . . .	6
3.1.1	Logging . . . . .	6
3.1.2	Metriken . . . . .	6
3.1.3	Tracing . . . . .	6
3.1.4	Fehlerberichte . . . . .	7
3.2	Praktiken aus der Fachpraxis . . . . .	7
3.3	Werkzeuge und Technologien . . . . .	7
3.4	Fachpraxis . . . . .	7
3.5	Literatur . . . . .	8
<b>4</b>	<b>Erstellung Proof-of-Concept</b>	<b>9</b>
4.1	Kriterien . . . . .	9
4.2	Vorstellung der Demoanwendung . . . . .	9

## *Inhaltsverzeichnis*

---

4.3	Konzept . . . . .	9
4.3.1	Architektur . . . . .	9
4.3.2	Datenverarbeitung . . . . .	9
4.3.2.1	Erhebung . . . . .	9
4.3.2.2	Auswertung . . . . .	10
4.3.2.3	Visualisierung . . . . .	10
4.3.3	Technologie-Stack . . . . .	10
4.3.4	Übertragbarkeit . . . . .	10
4.4	Implementierung . . . . .	10
<b>5</b>	<b>Ergebnis</b>	<b>11</b>
5.1	Demonstration . . . . .	11
5.2	Kriterien . . . . .	11
5.3	Übertragbarkeit . . . . .	11
5.4	Einschätzung von anderen Entwicklern (optional) . . . . .	11
<b>6</b>	<b>Abschluss</b>	<b>13</b>
6.1	Fazit . . . . .	13
6.2	Ausblick . . . . .	13
	<b>Anhang</b>	<b>14</b>
<b>7</b>	<b>Anhang</b>	<b>14</b>
	<b>Eidesstattliche Erklärung</b>	<b>15</b>
	<b>Abkürzungsverzeichnis</b>	<b>16</b>
	<b>Abbildungsverzeichnis</b>	<b>17</b>
	<b>Tabellenverzeichnis</b>	<b>18</b>
	<b>Quellcodeverzeichnis</b>	<b>18</b>
	<b>Literaturverzeichnis</b>	<b>19</b>

# 1 Einleitung

## 1.1 Motivation

Die Open Knowledge GmbH ist als branchenneutraler Softwaredienstleister in einer Vielzahl von Branchen wie Automotive, Logistik, Telekommunikation und Versicherungs- und Finanzwirtschaft aktiv. Zu den zahlreichen Kunden der Open Knowledge GmbH gehört auch ein führender deutscher Direktversicherer.

Ein Direktversicherer bietet Versicherungsprodukte seinen Kunden ausschließlich im Direktvertrieb, d. h. vor allem über das Internet und zusätzlich auch über Telefon, Fax oder B rief an. Im Unterschied zum klassischen Versicherer verfügt ein Direktversicherer jedoch über keinen Außendienst oder Geschäftsstellen, bei denen Kunden eine persönliche Beratung bekommen können. Da das Internet der primäre Vertriebskanal ist, gehört heute ein umfassender Online-Auftritt zum Standard. Dieser besteht typischerweise aus einem Kundenportal mit der Möglichkeit Angebote für Versicherungsprodukte berechnen und abschließen zu können, sowie persönliche Daten und Verträge einzusehen.

Während in der Vergangenheit Online-Auftritte i. d. R. als Webapplikation mit serverseitigen Rendering realisiert wurden, sind heutzutage Javascript-basierte Webapplikation mit clientseitigem Rendering die Norm. Bei einer solchen Webapplikation befindet sich die gesamte Logik mit Ausnahme der Berechnung des Angebots und der Verarbeitung der Antragsdaten im Browser des Nutzers.

Im produktiven Einsatz kommt es auch bei gut getesteten Webapplikationen hin und wieder vor, dass es zu unvorhergesehenen Fehlern in der Berechnung oder Verarbeitung kommen kann. Liegt die Ursache für den Fehler im Browser, z. B. aufgrund einer ungültigen Wertkombination, ist dies eine Herausforderung. Während bei Server-Anwendungen Fehlermeldungen in den Log-Dateien einzusehen sind, gibt es für den Betreiber der Anwendung i. d. R. keine Möglichkeit die notwendigen Informationen über den Nutzer und seine Umgebung abzurufen. Noch wichtiger ist, dass er mitbekommt, wenn ein Nutzer ein Problem bei der Bedienung der Anwendung hat. Ohne eine aktive Benachrichtigung durch den Nutzer, sowie detaillierte Informationen, ist es dem Betreiber nicht möglich, Kenntnis über das Problem zu erlangen, geschweige denn dieses nachzustellen.

Dies stellt ein Kernproblem von Webapplikationen dar [Fil20]. Im Rahmen der Arbeit soll daher ein Proof-of-Concept konzipiert und umgesetzt werden, welcher dieses Kernproblem am Beispiel einer Demoanwendung löst.

## 1.2 Zielsetzung

Das grundlegende Ziel dieser Arbeit soll es sein, den Betreibern einer JavaScript-basierten Webapplikation die Möglichkeit zu geben das Verhalten ihrer Applikation und die Interaktionen von Nutzern. Diese Nachvollziehbarkeit soll insbesondere bei Fehlerfällen u. Ä. gewährleistet sein, aber auch in sonstigen Fällen soll eine Nachvollziehbarkeit möglich sein. Eine vollständige Überwachung der Applikation und des Nutzers (wie bpsw. bei Werbe-Tracking) sind jedoch nicht vorgesehen. Daraus ergibt sich die Forschungsfrage:

Wie sieht ein Ansatz aus, um bei clientseitigen JavaScript-basierten Webapplikation den Betreibern eine Nachvollziehbarkeit zu gewährleisten?

Vom Leser wird eine Grundkenntnis der Informatik in Theorie oder Praxis erwartet, aber es sollen keine detaillierten Erfahrungen in der Webentwicklung vom Leser erwartet werden. Daher sind das Projektumfeld und seine besonderen Eigenschaften zu erläutern.

Die anzustrebende Lösung soll ein Proof-of-Concept sein, welches anhand einer bestehenden Demoanwendung erstellt werden soll. Die Demoanwendung soll repräsentativ eine abgespeckte JavaScript-basierte Webapplikation darstellen, bei der die zuvor benannten Hürden zur Nachvollziehbarkeit bestehen.

Vor der eigentlichen Lösungserstellung soll jedoch die theoretische Seite beleuchtet werden, indem die Nachvollziehbarkeit sowie Methoden und Praktiken zur Erreichung dieser beschrieben werden. Es gilt aktuelle Literatur und den Stand der Technik zu erörtern, in Bezug auf die Forschungsfrage. Beim Stand der Technik sollen Technologien aus Fachpraxis und Literatur näher betrachtet werden und beschrieben werden.

Weiterhin gilt es zu beleuchten, wie die Auswirkungen für die Nutzer der Webapplikation sind. Wurde die Leistung der Webapplikation beeinträchtigt (erhöhte Ladezeit, erhöhte Datenlast)? Werden mehr Daten von ihm erhoben und zu welchem Zweck?

Am Ende der Ausarbeitung soll überprüft werden, ob und wie die Forschungsfrage beantwortet wurde. Auch die Übertragbarkeit der erstellten Lösung (PoC) und Ergebnisse gilt es hierbei näher zu betrachten.

### 1.2.1 Abgrenzung

Die Demoanwendung wird als Single-Page-Application (SPA) realisiert, denn hier bewegt sich das Projektumfeld von der Open Knowledge GmbH. Bei der Betrachtung der Datenerhebung und -verarbeitung ist eine volle Konformität mit der DSGVO nicht zu prüfen.

Bei der Erörterung zum Stand der Technik sollen detaillierte Produktvorstellungen nicht das Ziel sein, auch ist keine umfassende Gegenüberstellung anzustreben.

### 1.3 Vorgehensweise

Zur Vorbereitung eines Proof-of-Concepts wird zunächst die Ausgangssituation geschildert. Speziell wird auf die Herausforderungen der Umgebung “Browser“ eingegangen, besonders in Hinblick auf die Verständnisk Gewinnung zu Interaktionen eines Nutzers und des Verhaltens der Applikation. Des Weiteren wird die Nachvollziehbarkeit als solche formal beschrieben und was sie im Projektumfeld genau bedeutet.

Darauf aufbauend werden allgemeine Methoden vorgestellt, mit der die Betreiber und Entwickler eine bessere Nachvollziehbarkeit erreichen können. Dabei werden die Besonderheiten der Umgebung beachtet und es wird erläutert, wie diese Methoden in der Umgebung zum Einsatz kommen können. Hiernach sind Ansätze aus der Literatur und Fachpraxis zu erörtern, welche eine praktische Realisierung der zuvor vorgestellten Methoden darstellen.

Auf Basis des detaillierten Verständnisses der Problemstellung und der Methoden wird nun ein Proof-of-Concept erstellt. Ziel soll dabei sein, die Nachvollziehbarkeit einer Webapplikation zu verbessern. Der Proof-of-Concept erfolgt auf Basis einer Demoanwendung, die im Rahmen dieser Arbeit erstellt wird.

Ist ein Proof-of-Concept nun erstellt, wird analysiert, welchen Einfluss es auf die Nachvollziehbarkeit hat und ob die gewünschten Ziele erreicht wurden (vgl. Zielsetzung).

### 1.4 Open Knowledge GmbH

Die Bachelorarbeit wird im Rahmen einer Werkstudententätigkeit innerhalb der Open Knowledge GmbH erstellt. Der Standortleiter des Standortes Essen, Dipl. Inf. Stephan Müller, übernimmt die Zweitbetreuung.

Die Open Knowledge GmbH ist ein branchenneutrales mittelständisches Dienstleistungsunternehmen mit dem Ziel bei der Analyse, Planung und Durchführung von Softwareprojekten zu unterstützen. Das Unternehmen wurde im Jahr 2000 in Oldenburg, dem Hauptsitz des Unternehmens, gegründet und beschäftigt heute 74 Mitarbeiter. Mitte 2017 wurde in Essen der zweite Standort eröffnet, an dem 13 Mitarbeiter angestellt sind.

Die Mitarbeiter von Open Knowledge übernehmen in Kundenprojekten Aufgaben bei der Analyse über die Projektziele und der aktuellen Ausgangssituationen, der Konzeption der geplanten Software, sowie der anschließenden Implementierung. Die erstellten Softwarelösungen stellen Individuallösungen dar und werden den Bedürfnissen der einzelnen Kunden entsprechend konzipiert und implementiert. Technisch liegt die Spezialisierung bei der Mobile- und bei der Java Enterprise Entwicklung, bei der stets moderne Technologien und Konzepte verwendet werden. Die Geschäftsführer als auch diverse Mitarbeiter der Open Knowledge GmbH sind als Redner auf Fachmessen wie der Javaland oder als Autoren in Fachzeitschriften wie dem Java Magazin vertreten.



## 2 Ausgangssituation

### 2.1 Browserumgebung

*Es gilt die generelle Umgebung und ihr Geschichte **kurz** zu umreißen.*

#### 2.1.1 Browserprodukte

*Hier soll kurz beschrieben werden, wie der aktuelle Browsermarkt aussieht und wie er sich entwickelt hat. Ob Cross-Browser-Inkompatibilitäten noch präsent sind, ist auch aufzuzeigen.*

#### 2.1.2 JavaScript

*Als Subgebiet des Browsers aber als eigentliche Hauptumgebung für die betrachteten Webapplikationen, soll JavaScript und dessen Integration im Browser betrachtet werden. Eine kurze geschichtliche Entwicklung und die Auswirkungen auf die heutige Welt der Webapplikationen ist zu erstellen.*

### 2.2 Clientbasierte Webapplikationen

#### 2.2.1 JavaScript-basierte Webapplikationen

*Als Einleitung zum nächsten Abschnitt sollen kurz JavaScript-basierte Webapplikationen erläutert werden. Es soll beschrieben werden, was sie von herkömmlichen Webapplikationen unterscheidet.*

#### 2.2.2 Single-Page-Applications

*Eine detailliertere Beschreibung zu SPAs soll nun erfolgen, da sie Hauptaugenmerk der Arbeit sind. Arbeitsweise, sowie Vorteile und Nachteile sollen für den Leser aufgezeigt und beschrieben werden.*

## 2.3 Nachvollziehbarkeit

*Zunächst soll dem Leser die Definition für Nachvollziehbarkeit erläutert werden, zudem ist ihr Bezug und ihre Relevanz in der Softwareentwicklung aufzuzeigen.*

### 2.3.1 Nutzen

*Hier gilt es ein echtes Beispiel für Nachvollziehbarkeit in einem Softwareprodukt aufzuzeigen, welches für den Leser verständlich und aufschlussreich ist.*

### 2.3.2 Nachvollziehbarkeit bei SPAs

*SPAs stellen aufgrund ihrer clientbasierten Natur eine besondere Herausforderung dar. Hier sollen allgemein die Herausforderungen von SPAs in Bezug auf die Nachvollziehbarkeit beschrieben werden.*

### 2.3.3 Browserbedingte Hürden

*Die Umgebung Browser besitzt, wie zuvor schon angerissen, einige besondere Hindernisse (CORS, Content-Security-Policy, Fernzugriff, etc.). Hier soll detailliert auf diese Hindernisse eingegangen werden und ihren Einfluss auf die Nachvollziehbarkeit.*

*Hier: Eine Überleitung zum nächsten Kapitel, in der Methoden zur besseren Nachvollziehbarkeit erklärt werden*

## 3 Methoden und Praktiken

*In diesem Kapitel soll beschrieben werden, wie eine Nachvollziehbarkeit in Webapplikationen erreicht werden kann. Spezielle Methoden und Praktiken sollen vorgestellt und beleuchtet werden.*

### 3.1 Methoden

#### 3.1.1 Logging

*Folgende Fragen sollen zur Methode beantwortet werden*

1. *Gibt es Besonderheiten zu Logging in anderen Projekten (Backend vs. Frontend)?*
2. *Wie können Logs an einen auswertenden Stakeholder gelangen?*
3. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

#### 3.1.2 Metriken

*Folgende Fragen sollen zur Methode beantwortet werden*

1. *Welche Metriken können definiert?*
2. *Wie können Metriken definiert werden?*
3. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

#### 3.1.3 Tracing

*Folgende Fragen sollen zur Methode beantwortet werden*

1. *Welche Nutzerinteraktionen sind zu tracen?*
2. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

### 3.1.4 Fehlerberichte

*Folgende Fragen sollen zur Methode beantwortet werden*

1. *Was genau sind Fehlerberichte (=Bug-Reports)*
2. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

### 3.2 Praktiken aus der Fachpraxis

*Neben den allgemeinen Methoden weisen einige Technologien in sich geschlossene Funktionsbereiche auf, die einen eigenen Ansatz darstellen. Diese Ansätze sollen hier getrennt von der jeweiligen Technologie beschrieben werden. (Beispielsweise "Real User Monitoring")*

### 3.3 Werkzeuge und Technologien

*Basierend auf dem Grundwissen über die Methoden und Praktiken, soll nun der Stand der Technik aus Fachpraxis und Literatur erörtert werden. Hierbei sollen Werkzeuge und Technologien und ihre Ansätze hervorgehoben werden und mit Hilfe welcher Methoden sie welches Ziel erreichen.*

*In diesem Abschnitt werden die Technologien vom Autor evaluiert und dann beschrieben. Es soll jedoch keine Gegenüberstellung erfolgen, sondern einfach eine Evaluierung, dieses Wissen soll dann später in der Erstellung des Lösungsansatzes helfen.*

*Weiterhin könnte beleuchtet werden, wie ähnliche Herausforderungen bei anderen „Fat-Client“-Lösungen (also nicht SPAs) angegangen werden, und ob man hier etwas lernen oder übertragen kann (und wenn nicht, warum nicht)?*

### 3.4 Fachpraxis

*Eine Menge (mindestens 3) an Technologien soll hierbei evaluiert werden, im Idealfall durch praktischen Einsatz.*

### 3.5 Literatur

1. *Gibt es hierzu Ansätze in der Literatur?*
2. *Wie sehen diese aus, welchen Zweck erfüllen sie?*
3. *Sind sie vergleichbar mit Ansätzen aus der Fachpraxis?*

## 4 Erstellung Proof-of-Concept

### 4.1 Kriterien

*Hier soll definiert werden, welche Kriterien der PoC erfüllen soll und wenn möglich sollen zusätzlich Möglichkeiten zur Überprüfung dieser Kriterien definiert werden.*

*Es ist zu beschreiben, wie diese Kriterien/Anforderungen zustande gekommen sind und welche Bedeutung sie für den Erfolg der Lösung besitzen.*

### 4.2 Vorstellung der Demoanwendung

*In diesem Abschnitt soll die Demoanwendung vorgestellt werden, anhand dessen das Proof-of-Concept erstellt wird. Damit das Proof-of-Concept erstellt werden kann, muss die Demoanwendung die zuvor beschriebenen Probleme aufweisen, hierbei sollen die Probleme möglichst realitätsnah sein und nicht frei erfunden.*

### 4.3 Konzept

#### 4.3.1 Architektur

*Hier soll die grobe Architektur geplant werden, welche Komponente es gibt und wie diese kommunizieren sollen.*

#### 4.3.2 Datenverarbeitung

##### 4.3.2.1 Erhebung

*Wie werden die Daten erhoben (Nennung der verwendeten Methoden!)? Wie gelangen die Daten an eine auswertende Komponente?*

#### 4.3.2.2 Auswertung

*Wie werden die Daten zusammengefasst und ausgewertet? Wie gelangt das Ergebnis an die darstellende Komponente?*

#### 4.3.2.3 Visualisierung

*Wie werden den Stakeholdern die Informationen präsentiert?*

#### 4.3.3 Technologie-Stack

*Welche Technologien werden eingesetzt, um die Architektur umzusetzen und zu erfüllen?*

#### 4.3.4 Übertragbarkeit

*Eignet sich das Konzept in Hinsicht auf Übertragbarkeit auf andere Softwareprojekte?*

*Dieser Punkt wird in Kapitel 5 erneut betrachtet und es wird versucht die tatsächliche Übertragbarkeit der erstellten Lösung zu erörtern.*

### 4.4 Implementierung

*Auf Basis des Konzeptes soll nun eine Implementierung erfolgen.*

## 5 Ergebnis

### 5.1 Demonstration

*Nach Abschluss der Implementierung, soll die Erweiterung auf nicht-technische Weise veranschaulicht werden. Hier soll dargestellt werden, wie die Nachvollziehbarkeit nun verbessert worden ist.*

### 5.2 Kriterien

*Die zuvor definierten Kriterien in 4.1 sollen hier überprüft werden.*

### 5.3 Übertragbarkeit

*Wie gut lassen sich die ermittelten Ergebnisse im PoC auf andere Projekte im selben Umfeld übertragen?*

### 5.4 Einschätzung von anderen Entwicklern (optional)

*Dieser Abschnitt kann ggf. wegfallen, wenn nicht genügend Zeit besteht oder der Nutzen nicht den Aufwand gerechtfertigt.*

*In diesem Abschnitt könnten Frontend-Entwicklern mit der Demo-Anwendung mit und ohne die Lösung konfrontiert werden. Daraufhin könnten diese befragt werden, ob TODO*

- 1. Wie gut entspricht die Demo-Anwendung einem realen Szenario?*
- 2. Sind die vorgestellten Probleme realitätsnah?*
- 3. Wie gut lassen sich die Probleme ohne die Lösung beheben?*
- 4. Wie gut lassen sich die Probleme mit der Lösung beheben?*



5. *Ist der Lösungsansatz zu komplex?*
6. *Gibt es Bedenken zum Lösungsansatz?*

## 6 Abschluss

### 6.1 Fazit

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

### 6.2 Ausblick

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

## 7 Anhang

# Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Dortmund, am .....  
(Unterschrift)

# Abkürzungs- und Erklärungsverzeichnis

Clientseitiges Rendering Der Server stellt dem Client lediglich die Logik und die notwendigen Daten bereit, die eigentliche Inhaltsgenerierung geschieht im Client. Für ein Beispiel siehe Unterabschnitt 2.2.2

PoC Proof-of-Concept

Serverseitiges Rendering Die darzustellenden Inhalte, werden beim Server generiert und der Client stellt diese dar. Beispielsweise sind Anwendungen mit PHP oder auch eine Java Web Application

# Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

# Literaturverzeichnis

- [Fil20] FILIPE, Ricardo Ângelo S.: *Client-Side Monitoring of Distributed Systems*, Universidade de Coimbra, Diss., 2020