

B

C

## Bachelorthesis

Konzipierung der Architektur einer Darstellungsschicht basierend auf aktuellen Webtechnologien im Hinblick auf eine zukünftige Anbindung an das Programm combit Relationship Manager

S



## Bachelorthesis

### Konzipierung der Architektur einer Darstellungsschicht basierend auf aktuellen Webtechnologien im Hinblick auf eine zukünftige Anbindung an das Programm combit Relationship Manager

von

**Jonas Reinwald**

Zur Erlangung des akademischen Grades eines

**Bachelor of Science (B.Sc.)**

an der Hochschule Konstanz für Technik, Wirtschaft und Gestaltung

Vorgelegt von:

Jonas Reinwald  
Wessenbergstr. 13  
78462 Konstanz  
293320

Erstbetreuer:

**Prof. Dr. Marko Boger**

Zweitbetreuer:

**Dipl.-Inf. Alexander Horak**  
combit GmbH  
Untere Laube 30  
78462 Konstanz

Ausgegeben am:

01.12.2018

Eingereicht am:

28.02.2019





# Inhaltsverzeichnis

<b>Danksagung</b>	<b>v</b>
<b>Zusammenfassung (Abstract)</b>	<b>vii</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Anforderungen</b>	<b>3</b>
<b>3 Frontend-Framework Vergleich</b>	<b>5</b>
<b>4 Entwicklung</b>	<b>7</b>
<b>5 Fazit</b>	<b>9</b>



# Danksagung

Stefanie Frischknecht: Ideen + Korrekturlesen

Prof. Dr. Marko Boger: Betreuer

Dipl.-Inf. Alexander Horak: Betreuer





# Zusammenfassung (Abstract)

Webapplikationen (mobile Apps, SPA, ... **FIXME: beispiel** **FIXME: formulierung**) machen heute bereits einen sehr großen Teil häufig benutzter (**FIXME: formulierung**) Software aus. In Zukunft wird der Anteil von webbasierter Software im Vergleich zu traditioneller Software noch weiter zunehmen (**FIXME: quelle**). Auch für Unternehmen ist es langfristig wichtig diesem Trend zu folgen um Kunden flexiblere Softwarelösungen anbieten zu können und nicht den Eindruck zu erwecken man könne nicht mit modernen Entwicklungen mithalten (**FIXME: formulierung**). Die Firma combit GmbH entwickelt mit dem combit Relationship Manager eine Software für das Management von Kundenbeziehungen (Customer Relationship Management), die zum momentanen Zeitpunkt aus einer Desktopapplikation, einer Webapplikation und einer mobile Webapplikation besteht, welche unabhängig voneinander entwickelt und benutzt werden. Im Rahmen dieser Bachelorthesis soll ein Architekturkonzept für eine einheitliche Oberfläche (Thin-Client, **FIXME: erklärung**) basierend auf moderner Webtechnologie erarbeitet werden, mithilfe derer die drei momentanen Ausführungen des combit Relationship Managers abgelöst werden können.

**FIXME: formulierung:** Zur Erstellung dieses Konzepts beschäftigt sich diese Arbeit unter anderem mit der Evaluierung eines geeigneten Frontend-Frameworks bzw. einer Frontend-Bibliothek, der Kommunikation zwischen neuem UI-Layer und Backend und der Unterstützung aller bisherigen Features der Desktopapplikation. Zwei besonders spannende Probleme existieren hier zum einen in der Flexibilität der momentanen Oberfläche, welche von jedem Nutzer individuell anpassbar ist. Diese Flexibilität soll durch eine automatische Konvertierung zu einem von der neuen UI darstellbarem Format und weiterhin anpassbarem Format erhalten bleiben. Zum anderen bietet die Desktopapplikation die Möglichkeit Skripte und automatisierte Aufgaben zur Anpassung der zugrunde liegenden Daten, für Darstellungsbedingungen und weitere Zwecke direkt auf dem Client auszuführen. Das Ausführen auf dem Client ist bei Webapplikationen nicht ohne Weiteres möglich, weshalb eine andere Lösung für diese Funktionalität gefunden werden muss.



# 1

## Einleitung

was (+ wieso web ansicht -> vorteile im vergleich zu bisherigem ansatz und inwiefern diese den aufwand rechtfertigen) uebersicht motivation + vorauszusehende schwierigkeiten?

- framework mit dem automatische umsetzung von UI (+ von kunden weiterhin möglichst einfach individualisierbar), entwicklung / wartung, testbarkeit möglichst einfach ist
- kommunikation zwischen UI und bisherigem Programm - anforderungen (scripting, filter, ...)



# 2

## Anforderungen



# 3

## Frontend-Framework Vergleich

komponenten in dli anschauen und uebersetzungen fuer alle uberlegen, dann beispielhaft von hand fuer verschiedene frameworks umsetzen und vergleichen. evtl. webaccess parser anschauen

chapter rename: technologie, zuesaetzlich: graphql vs rest, evtl. beide + weitere technologien





# 4

## Entwicklung



# 5

## Fazit