

# Arbeitstitel

## MASTER PROJEKT

für die Prüfung zum  
Master of Science  
des Studienganges Angewandte Informatik  
an der  
Fachhochschule Erfurt  
von  
**Sebastian Rieger**

Abgabedatum xx.xx.2017

|                               |                            |
|-------------------------------|----------------------------|
| Bearbeitungszeitraum          | 24 Wochen                  |
| Matrikelnummer                | 10286908                   |
| Ausbildungsfirma              | PDV Systeme<br>Erfurt      |
| Betreuer der Ausbildungsfirma | Dipl. -Inform. Nico Kaiser |
| Gutachter der Fachhochschule  | Prof. Rolf Kruse           |

## Erklärung

Ich, Sebastian Rieger, versichere hiermit, dass ich die vorliegende Bachelorarbeit mit dem Thema

Arbeitstitel

selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

---

Ort      Datum

---

Unterschrift

# Inhaltsverzeichnis

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Einleitung</b>                             | <b>4</b> |
| <b>2</b> | <b>Portalserver/ CMS-Systeme im Vergleich</b> | <b>5</b> |
| 2.1      | Typo3 . . . . .                               | 5        |
| 2.2      | Typo3 Neos . . . . .                          | 6        |
| 2.3      | Joomla . . . . .                              | 6        |
| 2.4      | Drupal . . . . .                              | 6        |
| 2.5      | Auswertung der möglichkeiten . . . . .        | 6        |
| <b>3</b> | <b>Typo3</b>                                  | <b>6</b> |
| 3.1      | Typo3 8.2 . . . . .                           | 6        |
| 3.2      | Extensions . . . . .                          | 6        |
| 3.3      | TypoScript . . . . .                          | 6        |
| 3.4      | Fluid . . . . .                               | 6        |
| <b>4</b> | <b>Modulare Extensions</b>                    | <b>6</b> |
| <b>5</b> | <b>Neue Webtechnologien</b>                   | <b>6</b> |
| 5.1      | Google Polymer . . . . .                      | 6        |
| 5.2      | AngularJS . . . . .                           | 6        |
| 5.3      | HTML5 . . . . .                               | 6        |
| 5.4      | CSS3 . . . . .                                | 6        |
| 5.4.1    | Bootstrap . . . . .                           | 6        |
| 5.4.2    | Foundation Framework . . . . .                | 6        |
| 5.5      | PHP7 . . . . .                                | 6        |
| 5.6      | Google Dart . . . . .                         | 6        |
| <b>6</b> | <b>Zusammenspiel der Technologien</b>         | <b>6</b> |
| <b>7</b> | <b>Zusammenfassung</b>                        | <b>6</b> |
| <b>8</b> | <b>Fazit</b>                                  | <b>6</b> |
| <b>9</b> | <b>Abkürzungsverzeichnis</b>                  | <b>7</b> |

# 1 Einleitung

In einer vernetzten Welt wie unserer, werden unablässig neue und bessere Web-Technologien entwickelt. Diese neuen Technologien bringen zum einen eine bessere Programmierfreundlichkeit mit sich, aber sie sind zum anderen durch neue Ansätze auch schneller als ältere Technologien.

Heutige Webanwendungen müssen eine Vielzahl an Kriterien erfüllen. Eine gute Webseite sollte heute möglichst auf einen Handy, sowie auf einem Desktop-PC oder Fernseher darstellbar sein. Um dies zu ermöglichen, müssen besondere Anforderungen an das Design gestellt werden. Natürlich muss sie auch von möglichst allen Browsern anzeigbar sein. Ein weiterer Punkt ist, dass Webseiten sich heute so flüssig und performant wie ein natives Programm verhalten sollen. Hierbei ist es natürlich auch wichtig vorhandene Hardware anzusprechen und mit ihr zu interagieren.

Das Ziel dieser Arbeit ist es herauszufinden, wie unter Zuhilfenahme moderner Technologien alle diese Kriterien möglichst gut erfüllt werden können. Es sollen Programmieransätze wie Google Polymer, AngularJS, HTML5, CSS3, PHP7 und Google Dart unter dem Portalserver vereint werden, welcher die Webseite beziehungsweise die Anwendung bereitstellt.

Es soll geprüft werden, wie und ob es möglich ist diese verschiedenen Technologien in möglichst modularen Portalserver-Erweiterungen unterzubringen, um eine möglichst performante und zu gleich leistungsstarke Webseite zu entwickeln. Dieses Oxymoron aufzulösen ist der Schwerpunkt der vorliegenden Arbeit, obwohl doch schon jetzt ersichtlich ist, dass ein auf beiden Seiten ein Kompromiss gefunden werden muss. Wie dieser genau aussieht, wird im weiteren Verlauf eine wichtige Rolle spielen.

Diese Arbeit soll der theoretischen Grundstock für eine weitere Arbeit sein, in der das Support-Portal der PDV System Erfurt GmbH neu entwickelt wird. Im Verlauf dieser Arbeit, wird evaluiert werden, welcher der gängigen Portalserver für ein Supportportal am besten geeignet ist. Außerdem wird geprüft werden, wie es möglich ist, modulare Erweiterungen für einen solchen Server zu entwickeln, um unnötige Datenredundanz zu vermeiden.

Durch einen modularen Aufbau, soll außerdem die Kommunikation zwischen verschiedenen Erweiterungen vereinfacht werden. Des Weiteren soll es möglich werden, nur beliebige Erweiterungen zu verwenden, wenn für den jeweiligen Use-Case nicht alle benötigt werden.

In den nun folgenden Abschnitten werden Programmierbeispiele und Hinweise gegeben, wie eine solche Neuentwicklung unter den Gesichtspunkten Performance, Umsetzbarkeit und Usability vorgenommen werden kann. Um Mitarbeitern und Kunden ein möglichst performantes Supportportal bieten zu können, welches den Arbeitsalltag eines jeden Nutzers erleichtert.

## 2 Portalserver/ CMS-Systeme im Vergleich

Das zukünftige Supportportal der PDV Systeme GmbH soll auf Basis eines Portalserver bzw. *Content Management System* (CMS)-Servers aufgebaut werden. Hierfür werden im folgenden einige Möglichkeiten genauer betrachtet.

Ein Portalserver, welcher für das Projekt heran gezogen wird muss die folgenden Eigenschaften aufweisen.

- Webseiten müssen frei gestaltbar sein
- Es muss die Möglichkeit bestehen Anwendungen für den Server zu entwickeln
- Das System muss Open-Source sein, um ggf. in den Quellcode eingreifen zu können
- Die Nutzer-Community sollte möglichst groß sein, damit Probleme leicht diskutiert und behoben werden können
- Der Server muss die Möglichkeit bieten Dateien zu verwalten, welche als Download oder Kontent in das Portal einfließen

Auf die Betrachtung reiner CMS-System wird an dieser Stelle verzichtet, da diese nicht die gewünschten Anforderungen eines Portalserver erfüllen. Ein Vergleich verschiedener reiner CMS-Systeme ist in der Bachelorarbeit “Konzept und prototypische Implementierung eines übergreifenden Dokumenten- und Medienmanagements” zu finden. [Rie15]

[Wik16]

### 2.1 Typo3

[Lob16]

## 2.2 Typo3 Neos

## 2.3 Joomla

## 2.4 Drupal

## 2.5 Auswertung der möglichkeiten

# 3 Typo3

## 3.1 Typo3 8.2

## 3.2 Extensions

## 3.3 TypoScript

## 3.4 Fluid

# 4 Modulare Extensions

# 5 Neue Webtechnologien

## 5.1 Google Polymer

## 5.2 AngularJS

## 5.3 HTML5

## 5.4 CSS3

### 5.4.1 Bootstrap

### 5.4.2 Foundation Framework

## 5.5 PHP7

## 5.6 Google Dart

# 6 Zusammenspiel der Technologien

# 7 Zusammenfassung

# 8 Fazit

## 9 Abkürzungsverzeichnis

**CMS** *Content Management System*

## Abbildungsverzeichnis



## Tabellenverzeichnis

## Literatur

- [Lob16] Parick Lobacher. *TYPO3 Extbase: Moderne Extensionentwicklung für TYPO3 CMS mit Extbase & Fluid*. CreateSpace Independent Publishing Platform, 2 edition, Februar 2016.
- [Rie15] Sebastian Rieger. Konzept und prototypische Implementierung eines übergreifenden Dokumenten- und Medienmanagements. Technical report, Karlsruher Institut für Technologie, August 2015.
- [Wik16] TYPO3. <https://de.wikipedia.org/wiki/TYP03>, September 2016.