



Masterstudiengang Informatik und Kommunikationssysteme (Vertiefung Informatik)  
Studiengang: MIKS3-INF

Hochschule Merseburg

# Evaluierung einer Dokumentenmanagementsoftware am Beispiel Alfresco

## Masterthesis

<b>Autor:</b>	Kevin Wagner
<b>geboren am:</b>	26. September 1994
<b>Praxisbetrieb:</b>	GISA GmbH
<b>Hochschulbetreuer:</b>	Prof. Dr. rer. pol. Uwe Schröter
<b>betrieblicher Betreuer:</b>	Hr. M. Sc. Matthias Friedrich
<b>Abgabetermin:</b>	01. September 2019
<b>Version vom:</b>	28. August 2019

---

Prof. Dr. rer. pol. Uwe Schröter  
Hochschulbetreuer

---

Kevin Wagner  
Student

## **Aufgabenstellung der Master-Thesis (M. Eng.)**

### **Thema**

Deutsch: Evaluierung einer Dokumentenmanagementsoftware am Beispiel Alfresco

Englisch: Evaluation of a document management software using the example Alfresco

### **Aufgabenstellung**

Das Thema wird in Kooperation mit der GISA GmbH mit Hauptsitz in Halle (Saale) bearbeitet. Ein Kunde dieser Firma hat die GISA GmbH beauftragt, eine Dokumentenmanagementsoftware bereitzustellen, die bestimmten Anforderungen gerecht werden muss. In dieser Arbeit soll untersucht werden, inwiefern die Software Alfresco für den geforderten Einsatzzweck geeignet und ob es möglich ist, ggf. fehlende Features nachzurüsten. Es soll ein Plan erstellt werden, wie die Software einzurichten und an die spezifischen Kundenanforderungen anzupassen ist.

### **Teilaufgaben, Arbeitsschritte**

- Definition Kundenanforderungen
- Problembeschreibung
- Sondierung Grundfähigkeiten Alfresco
- Eruiierung Bereitstellung weiterer benötigter Funktionen

Abzugebende Exemplare: 2

## Danksagung

Diese Stelle möchte ich nutzen, um mich bei all denjenigen zu bedanken, die mich bei der Anfertigung dieser Arbeit unterstützt haben.

Besonders hervorheben möchte ich meinen betrieblichen Betreuer Herrn Matthias Friedrich, welcher mir stets wertvolle Hinweise und fundiertes Hintergrundwissen zur Verfügung gestellt hat. Ich bedanke mich bei ihm für die exzellente Betreuung und fachliche Bereitschaft während der gesamten Bearbeitungszeit. Des Weiteren danke ich meinem Hochschulbetreuer Prof. Dr. rer. pol. Uwe Schröter für die Betreuung als studentischen Erstbetreuer und seine investierte Zeit.

Nicht zuletzt gilt mein besonderer Dank der Person, die meine berufliche Laufbahn erst möglich gemacht hat - meiner Mama.

# I Inhaltsverzeichnis

<b>Aufgabenstellung der Master-Thesis (M. Eng.)</b>	<b>II</b>
<b>Danksagung</b>	<b>III</b>
<b>II Abbildungsverzeichnis</b>	<b>VI</b>
<b>III Tabellenverzeichnis</b>	<b>VII</b>
<b>IV Quelltexte</b>	<b>VIII</b>
<b>V Abkürzungsverzeichnis</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Thematische Vorstellung . . . . .	1
1.2 Definition Dokumentenmanagementsystem . . . . .	1
1.3 Problemstellung und Ziele . . . . .	4
1.4 Vorstellung Alfresco . . . . .	5
1.4.1 Lizenz . . . . .	7
1.4.2 Preis . . . . .	8
1.5 Kosten und Nutzen . . . . .	10
<b>2 Technische Umsetzung</b>	<b>12</b>
2.1 Installation und Einrichtung . . . . .	12
2.1.1 Erreichbarkeit über HTTPS . . . . .	17
2.1.2 Persistierung der Daten . . . . .	21
2.2 Addons . . . . .	24
2.3 Content Repository . . . . .	26
2.4 Interaktion mit anderen Systemen . . . . .	26
2.4.1 Anbindung an KeyCloak . . . . .	27
2.4.2 Anbindung an inubit . . . . .	30
2.5 Corporate Design . . . . .	32
<b>3 Fachliche Umsetzung</b>	<b>33</b>
3.1 Regeln . . . . .	33
3.2 Zugriffskontrolle Dokumente . . . . .	34
3.3 Sicherung der Alfresco-Daten . . . . .	36
3.3.1 Replikation . . . . .	37
3.3.2 Backup . . . . .	38
<b>4 Zusammenfassung</b>	<b>39</b>

<b>Glossar</b>	<b>41</b>
<b>Anhang</b>	<b>43</b>
Anhang A: Freigabeprozess inubit und Alfresco (technisch) . . . . .	43
Anhang B: Vollständiges docker-dompose File mit gesamter Alfresco-Umgebung . . . . .	44
<b>Literatur</b>	<b>49</b>
<b>Selbstständigkeitserklärung</b>	<b>53</b>

## II Abbildungsverzeichnis

Abb. 1	Typische Einsparung bei Transport und Zugriffszeit von Papier- und elektronischen Dokumenten . . . . .	4
Abb. 2	Aufbau der Software Alfresco . . . . .	6
Abb. 3	Lizenzupload in Alfresco Share . . . . .	8
Abb. 4	Docker Logo . . . . .	12
Abb. 5	Systemdiagramm des Webservers und seiner Komponenten . . . . .	18
Abb. 6	Startseite Alfresco Share . . . . .	23
Abb. 7	Zusammenspiel von Content, Protokollen und Clients . . . . .	26
Abb. 8	Alfresco Realm in KeyCloak importieren . . . . .	28
Abb. 9	Maske zum Passwort ändern für Alfresco- und Keycloak-Nutzer . . . . .	29
Abb. 10	Systemdiagramm: Zusammenspiel Alfresco und inubit . . . . .	30
Abb. 11	Freigabeprozess inubit und Alfresco (fachlich) . . . . .	31
Abb. 12	Styleauswahl in Alfresco . . . . .	32
Abb. 13	Maske zur Erstellung einer Regel in Alfresco . . . . .	33
Abb. 14	Der Papierkorb in Alfresco auf der Profilseite. . . . .	36
Abb. 15	Maske zum Anlegen eines Replikationsjobs . . . . .	37
Abb. 16	Freigabeprozess inubit und Alfresco (technisch) . . . . .	43

### III Tabellenverzeichnis

Tab. 1	Einsparpotentiale durch Dokumentenmanagementsysteme . . . . .	3
Tab. 2	Alfresco Lizenzmodelle . . . . .	9
Tab. 3	Erreichbarkeit der Alfresco Module . . . . .	17

## IV Quelltexte

Qt. 1	Minimalbeispiel der Installation und des Starts eines Apache2 Webservers . . . . .	13
Qt. 2	Beispiel einer docker-compose.yml . . . . .	14
Qt. 3	Initialisierung des Docker-Setups von Alfresco . . . . .	15
Qt. 4	Anstarten der Alfresco Docker-Container . . . . .	16
Qt. 5	Logs beim Start der Alfresco-Container . . . . .	16
Qt. 6	nginx-Konfiguration als Proxy für Alfresco . . . . .	18
Qt. 7	Fehlermeldung aufgrund fehlgeschlagener CSRF-Prüfung . . . . .	19
Qt. 8	Beginn des CSRF-Knotens in der Konfigurationsdatei share-config-custom.xml	20
Qt. 9	Alfresco Share Konfiguration für share-config-custom.xml . . . . .	20
Qt. 10	Kopieren der originalen Konfigurationsdateien aus einem laufenden Container	21
Qt. 11	Ausgeben des Inhalts der share-security-config.xml . . . . .	21
Qt. 12	Dockerfile für Alfresco Share . . . . .	21
Qt. 13	Auszug aus docker-compose.yml mit Konfiguration der Volumes . . . . .	22
Qt. 14	Erweiterung der Umgebungsvariablen des Containers alfresco . . . . .	28
Qt. 15	Vollständiges docker-dompose File mit gesamter Alfresco-Umgebung . . . . .	44

## V Abkürzungsverzeichnis

ajp	Apache JServ Protocol
CD	Corporate Design
DMS	Dokumenten Management System
inubit	integrating your business and IT
JMX	Java Management Extensions
OCR	Optical Character Recognition
PID	Process Identifier

# 1 Einleitung

## 1.1 Thematische Vorstellung

Noch bis in die Mitte des ersten Jahrzehnts der 2000er Jahre wurden in vielen Unternehmen die meisten Daten auf Papier gedruckt und via Fax ausgetauscht. Blätter wurden in Aktenordnern sortiert sowie archiviert und nahmen fortan wertvollen Platz in Büroräumen ein. Musste dann ein Dokument gesucht werden, so war eine langwierige Suche nach dem richtigen Papier nicht selten. Eine komplizierte und fehleranfällige Methode, Daten zu verwalten und zu lagern.

Mit der voranschreitenden Digitalisierung änderte sich dies. Daten werden zunehmend in digitalen Dokumenten gehalten und archiviert. Diese Neuerung brachte aber gleichzeitig auch neuartige Probleme mit sich. So mussten fortan Dokumente zunehmend versioniert und der digitale Zugriff eingeschränkt und überwacht werden. Mit zunehmenden Möglichkeiten wuchsen auch die Anforderungen an ein digitales Management. Eine Software muss anpassungsfähig und funktional sein, um den aktuellen Anforderungen der Zeit gerecht zu werden. Immer mehr Aufgaben sollen automatisiert ausgeführt und Prozesse überwacht werden. Eine genaue Betrachtung einer Software ist also notwendig, um die Zweckmäßigkeit und Sinnhaftigkeit zu prüfen.

Für ein Projekt in der GISA GmbH soll eine Software zum Management von Dokumenten und zur Erstellung und Bearbeitung von Workflows bei einem Kunden eingeführt werden. Der Auftraggeber wünscht sich eine Umsetzung des Projektes mittels der Software *Alfresco*. Aus dem Auftrag ergeben sich diverse Anforderungen und Herausforderungen, die in Kapitel 1.3 beschrieben werden.

## 1.2 Definition Dokumentenmanagementsystem

*Dokument* bezeichnet in dieser Arbeit eine digitale Datei, die Inhalte persistiert. Dazu gehören Word-Dokumente, Excel-Tabellen, Powerpoint-Präsentationen, PDF-Dokumente und viele weitere Dateitypen, die Textdaten oder andere Datentypen enthalten. Nicht gemeint sind

physische Datenträger wie DVDs oder Magnetbänder.

Ein *Dokumentenmanagementsystem* (nachfolgend DMS genannt) ist ein Hilfsmittel, das die Verarbeitung von Dokumenten erleichtert und bestimmte Aufgaben übernimmt. Dazu gehören unter anderem

- die Erfassung von analogen Dokumenten und Aufarbeitung in eine geeignete elektronische Form,
- die Ablage und Speicherung dieser Dokumente in geeigneten Formaten,
- die Recherchefunktion nach Dokumenten und Inhalten im Bestand, sowie der Zugriff darauf,
- eine Bildschirmdarstellung, das Drucken und die Weiterleitung von abgerufenen Dokumenten an andere Kommunikationsverfahren wie Fax oder E-Mail und
- die Verteilung von Dokumenten.<sup>1</sup>

Ein DMS sollte zudem auch Schnittstellen zu anderen Applikationen besitzen.<sup>2</sup> Dadurch fügt es sich besser in die bestehende und zukünftige Systemlandschaft ein und bietet dem Nutzer eine größere Möglichkeit der Interaktion. Durch bessere Interaktionsmöglichkeiten und leichteren Zugriff erhöht sich zudem die Mitarbeitermotivation. Es werden Funktionen bereitgestellt, die Arbeiten einfacher und komfortabler machen.

Durch die Digitalisierung und Automatisierung von Dokumenten und Verwaltung ergibt sich eine stellenweise hohe Ersparnis verglichen mit Papierderivaten. Je nach verwendetem DMS und bisherigen Strukturen im Unternehmen fallen sie unterschiedlich aus. Im Allgemeinen kann jedoch von folgenden Zahlen ausgegangen werden:

<i><b>Einsparung in %</b></i>	<i><b>Einsparung</b></i>
50 - 90	<b>Kürzere Zugriffs- und Ablagezeiten</b> Transport- und Verweilzeiten entfallen weitgehend.

<sup>1</sup>Quelle: vgl. [GSSZ99], Seite 12

<sup>2</sup>Quelle: vgl. [Jeg02]

80 - 90	<b>kürzere Transportzeiten</b> große Einsparung, da Dokumente elektronisch transportiert werden
70 - 95	<b>kürzere Suchzeiten</b> sehr hohe Zeitersparnis durch elektronische Suche
10 - 25	<b>Verbesserung der Produktivität</b> Personal kann kontinuierlich arbeiten und wird nicht durch Suche nach Dokumenten unterbrochen
10 - 35	<b>Personaleinsparung</b> ergibt sich durch gesteigerte Produktivität und schnellere Bearbeitungszeit
10 - 55	<b>Reduzierung der Bürosystemkosten</b> durch Einsparung von Aktenschränken, Ordnern usw.
20 - 55	<b>Reduzierung der notwendigen Formulare und Dokumente,</b> die als Papier für interne Zwecke erzeugt werden müssten.
30 - 55	<b>Reduzierung des Platzbedarfs</b> Die Lagerung benötigt weniger Platz, da digitale Speichermedien eine weitaus höhere Speicherdichte haben.

Tabelle 1: Einsparpotentiale durch Dokumentenmanagementsysteme <sup>3</sup>

Zudem gibt es noch weitere Vorteile, deren monetärer Nutzen jedoch schwer einzuschätzen ist. Dazu zählt unter anderem die einfache Replizierbarkeit von Dokumenten auf beliebige Datenträger, die dem klassischen Kopieren von Dokumenten gegenübersteht. Am Beispiel *Kopiervorgang* wird deutlich, dass ein Dokument schnell und unbeabsichtigt durch den Kopierer verfälscht werden kann. Bei einem elektronischen Kopiervorgang werden diese Fehler größtenteils ausgeschlossen. Außerdem können Dokumente schnell und einfach zur Verfügung

---

<sup>3</sup>Quelle: vgl. [GSSZ99], Seite 13

gestellt werden.<sup>4</sup>

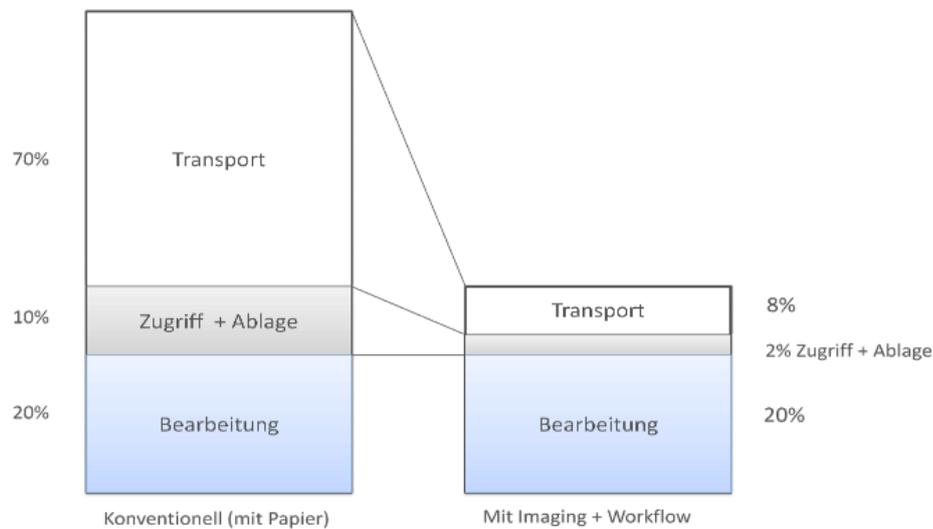


Abbildung 1: Typische Einsparung bei Transport und Zugriffszeit von Papier- (links) und elektronischen Dokumenten (rechts)<sup>5</sup>

Die Abbildung 1 zeigt deutliche Einsparungen bei dem Transport der Dokumente. Lange Übertragungswege zwischen Standorten via Post entfallen. Stattdessen können Dokumente über eine zentrale Onlineplattform ausgetauscht und gesucht werden, was Zugriffszeiten deutlich minimiert.

### 1.3 Problemstellung und Ziele

Eine Kundenvoraussetzung ist, dass das DMS *Alfresco* nicht autark betrieben, sondern an die bereits vorhandene Software *KeyCloak* gekoppelt werden soll. Die kostenfreie Nutzerwaltungssoftware *KeyCloak* verwaltet User und Passwörter in einem separatem System. Die Nutzerdaten sollen aus dem System übernommen und stetig synchronisiert werden. Weiterhin soll *Alfresco* an die Prozessierungssoftware *inubit* gekoppelt werden. Dabei sollen aus beliebigen Vorgängen die jeweiligen Freigabeprozesse herausgelöst und in *Alfresco* übertragen werden.

Außerdem muss die Software um Funktionen, die vom Kunden gefordert, aber nicht vorhan-

<sup>4</sup>Quelle: vgl. [Jeg02], Seite 39

<sup>5</sup>Quelle: [Jeg02], Seite 40

den sind, erweitert werden. Dafür ist ein sinnvolles Vorgehen zu finden, damit die Software auch zukunftssicher betrieben und aktuell gehalten werden kann. Abschließend soll untersucht werden, welche Anforderungen die Software out-of-the-box erfüllt.

Im Rahmen dieser Arbeit soll beispielhaft eine lauffähige Anwendungslandschaft aufgezeigt werden. Dies soll unter Zuhilfenahme von *Docker*, einer Virtualisierungsumgebung, geschehen, die in Kapitel 2.1 erläutert wird. Außerdem muss die Umgebung vollständig reproduzierbar sein.

Es soll dem Nutzer möglich sein, Dokumente in einer vorgegebenen Struktur zentral zu speichern, zu verwalten, zu bearbeiten und abzurufen. Der Zugriff soll durch bereits vorhandene Schutzmaßnahmen des bestehenden Firmennetzes, wie beispielsweise Firewalls, gesichert werden. Daten sollen durch autorisierte Personen zu jedem Zeitpunkt abrufbar sein.

Es ist kein Ziel dieser Arbeit, Daten zu migrieren oder eine vollständig lauffähige Software bereitzustellen.

## 1.4 Vorstellung Alfresco

Die vom Kunden vorgeschlagene Software *Alfresco* wird seit 2005<sup>6</sup> entwickelt und besteht aus mehreren Modulen. Sie ist den *Enterprise Content Management* Systemen zuzuordnen.<sup>7</sup> Diese Arbeit konzentriert sich nur auf den Teil der Dokumentenverwaltung. Geschrieben wurde die Software in *Java*, *JavaScript*, *JavaServer Pages* und *PHP*<sup>8</sup>. Die Plattform ist OpenSource<sup>9</sup>, modern und sicher.<sup>10</sup>

Den Kern der Software bildet ein Repository<sup>11</sup>-System, das die Dokumente verwaltet, Inhalte, Metadaten und Volltext-Indizes persistiert. Über Schnittstellen in verschiedenen Sprachen können Entwickler eigene Funktionen hinzufügen oder andere Softwares anbinden.

---

<sup>6</sup>Quelle: vgl. [Alf06]

<sup>7</sup>Quelle: vgl. [Alfg]

<sup>8</sup>Quelle: vgl. [rwe05]

<sup>9</sup>Begriffserklärung s. Glossar auf Seite 41

<sup>10</sup>Quelle: vgl. [For19]

<sup>11</sup>Begriffserklärung s. Glossar auf Seite 41

Abbildung 2: Aufbau der Software Alfresco<sup>12</sup>

*Alfresco* ist in Java geschrieben und kann somit auf nahezu jeder Plattform ausgeführt werden. Unterstützt wird es durch das *Spring Framework*<sup>13</sup>, welches die Möglichkeit bereitstellt, Funktionen in Module auszulagern. Dazu gehören Versionierungen, Sicherheitsfunktionen und Regelmanagement.<sup>14</sup> Im Folgenden werden die wichtigsten Module kurz benannt und erläutert.

**Activiti** ist ein freies Workflow-Management-System, in dem Business-Prozesse modelliert und ausgeführt werden können. Dieses ist in der freien Version von Alfresco implementiert, in der kostenpflichtigen Enterprise Version wird es durch die Komponente **Alfresco Process Services** erweitert.<sup>15</sup>

Alfresco **Share** ist das Herzstück der Software. In diesem Modul werden Dokumente gespeichert und verwaltet. Dies geschieht in sog. *Sites*, was mit einem Projekt zu vergleichen ist. Zu jeder Site können zudem Extras wie ein Blog, Kalender, Wiki usw. angelegt werden.

Apache **solr** „basiert auf der Java-Bibliothek von *Apache Lucene* und stellt mächtige Suchfunk-

<sup>12</sup>Quelle: [Alfe]

<sup>13</sup>Begriffserklärung s. Glossar auf Seite 41

<sup>14</sup>Quelle: vgl. [Alfe]

<sup>15</sup>Quelle: vgl. [Alfg]

tionen zur Verfügung, die sich auch im Big-Data-Umfeld einsetzen lassen.“<sup>16</sup> Folglich übernimmt solr hier also die Indexierung und Suche nach Dokumenten, Metadaten, Inhalten, sowie anderen Eigenschaften.

### 1.4.1 Lizenz

Alfresco wird in zwei Versionen angeboten: Die kostenfreie *Community Edition* und die für Firmen gedachte *Enterprise Edition*.

Die freie Version unterliegt der *GNU LGPL*-Lizenz. Der Unterschied zur bekannteren *GNU GPL*-Lizenz besteht darin, dass bei der erweiterten Variante die lizenzierten Produkte auch in proprietärer und nicht nur freier Software verwendet werden dürfen.<sup>17</sup> Die kostenpflichtige Variante von Alfresco ist unter einer „handelsüblichen“ Lizenz lizenziert.<sup>18</sup> Außerdem wird der Preis individuell pro Nutzer und Server abgerechnet.<sup>19</sup>

Der Hauptunterschied liegt im verfügbaren Support. Während die Community Version nur das öffentliche Forum als Unterstützung besitzt, können Kunden mit einer bezahlten Lizenz den Hersteller bei Problemen direkt kontaktieren.

Die funktionalen Unterschiede dagegen sind gering. Neue Features werden zunächst in der Community Version veröffentlicht und durch die Vielzahl der Nutzer getestet. Erst wenn eine gewisse Zeit keine Fehler gemeldet wurden, werden die Funktionen in die Enterprise Edition überführt. Es wird also eine höhere Stabilität geboten als in der freien Version.<sup>20</sup> Zusätzlich gibt es neue Funktionen, die nur in der Bezahlversion verfügbar sind. Dies sind:

- Skalierbarkeit und Leistungsoptimierungen wie Clustering<sup>21</sup> und JMX<sup>22</sup> Administration
- Synchronisierungen zwischen Instanzen auf eigener Hardware und Cloud Services

---

<sup>16</sup>Quelle: [Ste06]

<sup>17</sup>Quelle: vgl. [Fre17]

<sup>18</sup>Quelle: [res06]

<sup>19</sup>Quelle: vgl. [Alfa]

<sup>20</sup>Quelle: [Alf19a]

<sup>21</sup>Begriffserklärung s. Glossar auf Seite 41

<sup>22</sup>Begriffserklärung s. Glossar auf Seite 41

- Integration in proprietäre Softwares wie *Outlook*, *Microsoft Office* und *Kofax*
- Mehr Einsatzmöglichkeiten für proprietäre Datenbankmanagementsysteme wie *MySQL* und *Oracle DB* <sup>23</sup>

Die freie Version benötigt keine Lizenz. Die Demo der Bezahlversion schaltet nach zwei Tagen in einen read-only Modus, in dem im System nichts mehr verändert werden kann. Es muss zunächst eine Lizenz beantragt und eingespielt werden. Dazu wählt man in der *Administrator Console* im Unterpunkt *Lizenz* die erhaltene Lizenz aus und lädt sie hoch, so wie in Abbildung 3 zu sehen.

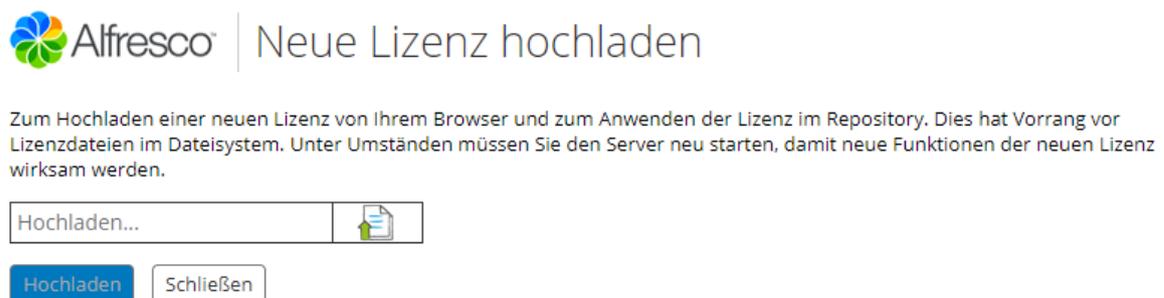


Abbildung 3: Lizenzupload in *Alfresco Share*

Eine allgemeine Aussage, welcher Plan wann genutzt werden soll, kann nicht getroffen werden. Eine Bewertung ist in jedem Einzelfall anhand der Anforderungen und Umstände notwendig. Um im betroffenen Projekt eine Entscheidung treffen zu können, müssen zunächst die verfügbaren Lizenzpläne untersucht werden.

### 1.4.2 Preis

Genaue Preise werden vom Hersteller von Alfresco nicht genannt, es muss für jeden Lizenzplan ein Kontaktformular ausgefüllt und ein individuelles Angebot eingeholt werden. Für die Enterprise Edition werden drei verschiedene Lizenzmodelle angeboten, die in Tabelle 2 beschrieben sind.

---

<sup>23</sup>Quelle: vgl. [Alfa]

<b>Plan</b> <b>Eigenschaft</b>	<b><i>Starter</i></b>	<b><i>Business</i></b>	<b><i>Enterprise</i></b>
Anzahl Nutzer	bis zu 100	bis zu 300	bis zu 1.000
Anzahl nutzbare CPU-Kerne	4	4	8 + Clustering
Support	Während der Geschäftszeiten	Während der Geschäftszeiten	24/7
zusätzliche Module	<ul style="list-style-type: none"> <li>• Microsoft Outlook Client</li> <li>• Alfresco Records Management</li> <li>• Alfresco Content Connector für Salesforce.com</li> <li>• Amazon S3 Connector</li> </ul>	Wie <i>Starter</i> , plus <ul style="list-style-type: none"> <li>• Alfresco Media Management</li> <li>• Document Transformation Engine</li> <li>• Media Transformation Engine</li> <li>• EMC Centera Connector</li> <li>• Alfresco System Receiver</li> </ul>	Wie <i>Business Plan</i> , plus <ul style="list-style-type: none"> <li>• Alfresco Index Engine</li> <li>• Alfresco Content Encryption</li> </ul>

Tabelle 2: Alfresco Lizenzmodelle <sup>24</sup>

Der Kunde erwartet in etwa 2000 Dienstleister, die jeweils einen eigenen Nutzer in Alfresco benötigen. Daher wären die Lizenzpläne *Starter* und *Business* nicht nutzbar. Zusätzliche Mo-

---

<sup>24</sup>Quelle: vgl. [alf19]

dule, wie beispielsweise die Outlook-Integration werden weder vom Kunden noch von dessen Dienstleistern benötigt. Der einzig deutliche Vorteil einer Bezahlversion im Vergleich zu der kostenfreien Variante ist der Support bei auftretenden Problemen durch den Hersteller. Unter dem Gesichtspunkt der Kundenanforderungen wurde die Entscheidung getroffen, die kostenfreie Version für das Projekt zu verwenden.

## 1.5 Kosten und Nutzen

Der Betrieb einer Software verursacht Kosten. Dazu zählen Kosten für die initiale Installation, den Betrieb und die Wartung. Bevor der Aufwand betrieben wird, ein Dokumentenmanagementsystem einzuführen und zu unterhalten, muss zunächst geprüft werden, ob ein solcher Aufwand überhaupt gerechtfertigt ist und die Kosten der gegenwärtigen Dokumentenhaltung nicht übersteigt. Weiterhin muss eine Wartung der Software in Zukunft sichergestellt werden.

*Holger Reibold* schreibt in seinem Buch „Alfresco 5.0 kompakt“, dass es eine „nennenswerte Stückzahl“ an Dokumenten bedarf, um ein DMS einsetzen zu können.<sup>25</sup> Er spezifiziert diese aber nicht genauer.

Die Softwareentwicklungsfirma *aimprosoft* beschreibt in ihrem Blog fünf Gründe zur Benutzung eines DMS:

### 1. Wiederherstellungsmöglichkeiten

Ein einmal zerstörtes Dokument aus Papier ist für immer verloren. Eine digitale Kopie kann aus Backups wiederhergestellt werden.

### 2. Mehr Ordnung durch weniger Papier

Der Platzverbrauch durch ausgedruckte Dokumente ist enorm. Eine digitale Speicherung schafft mehr Platz im Archiv, aber auch auf Schreibtischen.

---

<sup>25</sup>Quelle: [Hol15], Seite 9

### 3. **Zusammenarbeit verbessert sich**

Es ist nahezu unmöglich, dass mehrere Personen an einem Stück Papier arbeiten. Unter Zuhilfenahme eines DMS ist dies jedoch machbar.

### 4. **Datensicherheit hat eine hohe Priorität**

Eine Zugriffskontrolle durch eine Nutzerverwaltung ist sicherer als das Wegschließen in Archivräumen.

### 5. **Umweltschutz**

Dokumente, die nicht ausgedruckt werden, sparen Umweltressourcen und somit auch Verbrauchskosten.<sup>26</sup>

Eine genaue Prüfung der Umstände bei dem auftraggebenden Kunden hat ergeben, dass ein Einsatz einer Dokumentenmanagementsoftware sinnvoll ist. Dienstleister des Kunden sollen ihm regelmäßig mehrere Dokumente zukommen lassen. Bisher geschieht dies postalisch. Neben der zeitlichen Verzögerung sind vor allem anfallende Portokosten entscheidungsrelevant. Da eine sehr hohe Anzahl an Dienstleistern erwartet wird, verringert sich der Verwaltungsaufwand deutlich.

---

<sup>26</sup>Quelle: [Aim07]

## 2 Technische Umsetzung

### 2.1 Installation und Einrichtung

Die Software Alfresco kann auf verschiedenen Wegen installiert werden. Eine plattformunabhängige Variante ist die Installation via Docker.

Der Begriff *Docker* bezieht sich im Kontext dieser Masterarbeit auf die Virtualisierungsumgebung zur Ausführung von Containern. Ein Container ist vergleichbar mit einer leichtgewichtigen virtuellen Maschine, in der jeweils nur eine Anwendung zur Ausführung bereitsteht. Neben dieser Anwendung sind außerdem alle benötigten Abhängigkeiten der Software und die Minimalversion eines Betriebssystems im Container vorhanden. Dadurch entsteht eine Unabhängigkeit zu allen anderen Umgebungsvariablen des Systems, auf dem der Container ausgeführt werden soll.<sup>28</sup>

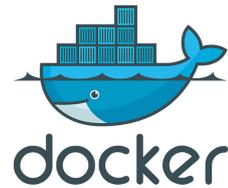


Abbildung 4: Docker Logo<sup>27</sup>

“ If it runs on your machine, it runs anywhere. ”

*Leitspruch Docker*<sup>29</sup>

Um zu definieren, wie ein solcher Container aufgebaut ist, muss ein sog. *Dockerfile* erstellt werden. In dieser Datei werden Formvorschriften und Befehle gesammelt, die in ihrer Gesamtheit ein sog. *Image* bilden. Ein Beispiel dafür ist in Quelltext 1 zu sehen. Aus einem Image wird dann durch Ausführung der Formvorschriften und Installation der Software ein *Container*. Dieser kann nun gestartet und damit die darin installierte Anwendung ausgeführt werden. Um mit der Anwendung zu interagieren, können Ports vom Hostsystem in den Container weitergeleitet werden. Darüber kann die Anwendung im Container angesprochen werden.

<sup>27</sup>Quelle: [Jac07]

<sup>28</sup>Quelle: vgl. [Red]

<sup>29</sup>Quelle: [Ges10]

```
1 FROM debian:9-slim
2
3 RUN apt-get update && \
4     apt-get install -y apache2
5
6 EXPOSE 80
7 CMD ["service apache2 start"]
```

Quelltext 1: Minimalbeispiel der Installation und des Starts eines Apache2 Webservers

Zum Austausch von Dateien können Verzeichnisse in sog. *Volumes* eingebunden werden. Dies ist vergleichbar mit einer Netzwerkfreigabe. Daten in einem Volume bleiben bei Neuerstellung eines Containers erhalten. Alle anderen Daten, die nicht in einem Volume liegen, werden bei Neuerstellung des Containers verworfen. Dies gilt auch für alle anderen Modifikationen, die während der Laufzeit am Container vorgenommen wurden. Beispielsweise nachträglich installierte oder deinstallierte Pakete werden bei Neuerstellung des Containers verworfen.

Ein Container wird solange ausgeführt, wie die *Process ID (PID)*<sup>30</sup> der Hauptanwendung existiert. Das bedeutet, dass ein Container solange läuft, wie die sich darin befindliche Anwendung ausgeführt wird. Wird beim Start des Containers, zum Beispiel durch einen Konfigurationsfehler, der Start der Anwendung verhindert, wird der Container sofort wieder gestoppt. Davon abhängige Container werden in diesem Fall nicht gestartet.

Um mehrere Container und zugehörige Parameter zu verwalten, empfiehlt es sich, das Programm *docker-compose* zu nutzen. Damit werden alle Container in einer *yml*-Datei erfasst und können mit einem selbst gewählten Namen angesprochen werden.

---

<sup>30</sup>Begriffserklärung s. Glossar auf Seite 41

```
1  version: '3'
2  services:
3    apache:
4      build: ./apache/
5      ports:
6        - 8080:80
7      volumes:
8        - ./data/www/:/var/www/
9      depends_on:
10       - mysql
11  mysql:
12    image: mysql
```

Quelltext 2: Beispiel einer docker-compose.yml

In dieser Datei werden Parameter, wie belegte Ports oder genutzte Volumes definiert. Die Werte links neben den Doppelpunkten beziehen sich dabei immer auf Referenzen zum Host-System, rechts der Doppelpunkte auf Referenzen in dem jeweiligen Container. So bedeutet beispielsweise die Port-Angabe `8080:80` in den Zeilen 5 und 6 des Quelltextes 2, dass der Host-Port `8080` auf den Port `80` im Container weiterleitet. Da Docker ein eigenes, virtuelles Netzwerk für die Container erstellt, können die Container untereinander kommunizieren. Soll ein Containerport nicht auf einen Hostport weiterleiten, sondern nur im internen Docker-Netzwerk veröffentlicht werden, so entfällt die Hostportangabe. Die Zeile verkürzt sich auf die Angabe `ports: 80`.

Darüber hinaus können über die Felder `links` und `depends_on` Startreihenfolgen der Container festgelegt werden. Ein Container, der von einem anderen abhängt, wird erst dann angestartet, wenn der andere Container vollständig und erfolgreich hochgefahren wurde. Durch das stufenweise Starten wird verhindert, dass ein Webserver eher startet als sein dazugehöriger Datenbankserver. Damit werden Fehlermeldungen unterdrückt und ein unerwartetes Verhalten vermieden.

Um den Installationsprozess über Docker zu vereinfachen, stellt der Hersteller für *Alfresco* ein vorgefertigtes Setup bereit.

```
1 #!/bin/bash
2
3 git clone https://github.com/Alfresco/acs-deployment.git
4 cd acs-deployment
5 git checkout support/HF/1.0
6
7 cd docker-compose/
8 docker-compose up -d
```

Quelltext 3: Initialisierung des Docker-Setups von Alfresco

Durch Ausführung der in Quelltext 3 ersichtlichen Befehle starten mehrere Container, die unterschiedliche Aufgaben übernehmen:

- **alfresco** - Alfresco Applikation  
Es gibt hier zwei verschiedene Varianten: Das Image *alfresco-content-repository* wird für die Enterprise-Variante genutzt, wohingegen für die kostenlose Variante das Image *alfresco-content-repository-community* nutzt.
- **alfresco-pdf-renderer** - Erzeugung von PDF-Vorschaubildern<sup>31</sup>
- **imagemagick** - Bildverarbeitung für die Dokumentenverwaltung<sup>32</sup>
- **libreoffice** - Office Integration in die Dokumentenverwaltung<sup>33</sup>
- **postgres** - Speicher für Content-Repository-Daten, wie z.B. Metadaten
- **share** - Dokumentenverwaltung

---

<sup>31</sup>Quelle: vgl. [Alf19c]

<sup>32</sup>Quelle: vgl. [Alf19d]

<sup>33</sup>Quelle: vgl. [Alf19e]

- **solr6** - OpenSource Suchplattform, dient zur Speicherung von Volltextindizes zur Suche von Dokumenten<sup>34</sup>

Wurden die Container erfolgreich gestartet, so erscheint folgende Ausgabe:

```
1 Creating network "docker-compose_default" with the default driver
2 ...
3 Creating docker-compose_alfresco-pdf-renderer_1 ... done
4 Creating docker-compose_solr6_1 ... done
5 Creating docker-compose_postgres_1 ... done
6 Creating docker-compose_imagemagick_1 ... done
7 Creating docker-compose_libreoffice_1 ... done
8 Creating docker-compose_share_1 ... done
9 Creating docker-compose_alfresco_1 ... done
10 Attaching to docker-compose_postgres_1, docker-compose_solr6_1, docker-
    compose_alfresco-pdf-renderer_1, docker-compose_imagemagick_1, docker-
    compose_alfresco_1, docker-compose_libreoffice_1, docker-
    compose_share_1
11 ...
```

#### Quelltext 4: Anstarten der Alfresco Docker-Container

Daraufhin führen die Anwendungen Initialisierungsprozesse durch. Dies kann einige Minuten dauern. Den erfolgreichen Abschluss erkennt man durch Ausführung des Befehls `docker-compose logs -f` und dem Warten auf die Ausgabe, dass der Startvorgang erfolgreich abgeschlossen wurde.

```
1 ...
2 alfresco_1 | 2018-08-28 08:40:30,386 INFO [management.subsystems.
    ChildApplicationContextFactory] [http-nio-8080-exec-10] Starting '
    Transformers' subsystem, ID: [Transformers, default]
```

<sup>34</sup>Quelle: vgl. [Hol15], Seite 15

```

3 alfresco_1 | 2018-08-28 08:40:30,661 INFO [management.subsystems.
  ChildApplicationContextFactory] [http-nio-8080-exec-10] Startup of
  'Transformers' subsystem, ID: [Transformers, default] complete

```

### Quelltext 5: Logs beim Start der Alfresco-Container

Ab diesem Zeitpunkt ist Alfresco Share unter der URL `http://localhost:8080/share/` erreichbar.<sup>35</sup>

#### 2.1.1 Erreichbarkeit über HTTPS

Es ist enorm wichtig, Kundendaten nicht unverschlüsselt über öffentliche Netze zu übertragen. Aus diesem Grund müssen alle Teile der Software über einen verschlüsselten HTTPS-Kanal erreichbar sein. Im Folgenden wird beschrieben, wie dies umgesetzt werden kann. Der Einrichtungsprozess für TLS-Zertifikate wird nicht erklärt, da dies vom Thema der Arbeit abweichen würde.

Durch die Installation via Docker sind die einzelnen Services in verschiedenen Containern über verschiedene Ports erreichbar. Konkret geht es um folgende Services und Adressen (der Hostanteil in den Adressen repräsentiert den Containernamen):

	<i>Service</i>	<i>URL</i>
Erreichbar über Browser	Alfresco Admin Console Share	<code>http://alfresco:8080/</code> <code>http://share:8080/</code>
Erreichbar über <i>ajp</i> <sup>3637</sup>	solr6	<code>ajp://solr6:8983/</code>

Tabelle 3: Erreichbarkeit der Alfresco Module

Es ist nicht nutzerfreundlich, wenn einzelne Services über ihre Ports angesprochen werden müssen. Zudem kann dies insbesondere in Unternehmen zu Konflikten mit Sicherheitsricht-

<sup>35</sup>Quelle: vgl. [Alf19b]

<sup>36</sup>Begriffserklärung s. Glossar auf Seite 41

<sup>37</sup>Das Protokoll *ajp* wird nur von den Webserver *Apache2* unterstützt. *nginx* unterstützt dieses Protokoll nicht.

linien führen, sollten unkonventionelle Ports über das Internet erreichbar sein. Es empfiehlt sich, Anfragen an Alfresco gebündelt an einem Proxyserver zu empfangen und je nach aufgerufener URI<sup>38</sup> an den jeweiligen Container weiterzuleiten. Die Kommunikation zwischen Endanwender und Proxyserver ist dabei verschlüsselt, die Kommunikation zwischen Proxy und Containern nicht. Dies ist aber auch nicht notwendig, da die Anfragen auf einem Host bleiben und nicht über ein Netzwerk geschickt werden.

Als Proxyserver fungiert im hier verwendeten Beispiel der Webserver *nginx*, der über das Modul *proxy\_pass* die Requests an die Container weiterleitet.

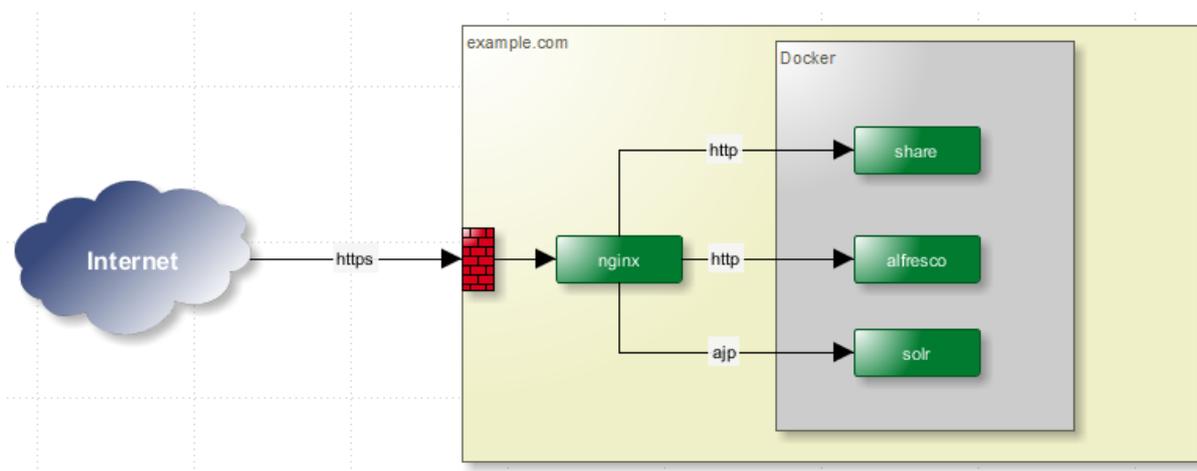


Abbildung 5: Systemdiagramm des Webservers und seiner Komponenten

```
1 location /alfresco/ {
2     proxy_pass http://alfresco:8080/alfresco/;
3 }
4 location /share/ {
5     add_header Access-Control-Allow-Origin *;
6     proxy_pass http://share:8080/share/;
7 }
8 location /solr/ {
9     proxy_pass http://solr6:8983/solr/;
```

<sup>38</sup>Begriffserklärung s. Glossar auf Seite 41

10 }

### Quelltext 6: nginx-Konfiguration als Proxy für Alfresco

Die Container *alfresco* und *solr6* können nun bereits über den Proxy kommunizieren, der Container *share* benötigt noch weitere Konfigurationen. Bei einem Login wird Alfresco diesen derzeit mit der in Quelltext 7 gezeigten Fehlermeldung zurückweisen. Dieser Fehler tritt dann auf, wenn die Weiterleitung vom Proxyserver nicht vollständig konfiguriert ist.

```
1 2019-05-27 12:06:41,713 ERROR [alfresco.web.site] [http-nio-8080-exec
-1] javax.servlet.ServletException: Possible CSRF attack noted when
asserting referer header 'https://example.com/share/page/'. Request:
POST /share/page/dologin, FAILED TEST: Assert referer POST /share/
page/dologin :: referer: 'https://example.com/share/page/' vs server
& context: http://example.com/ (string) or (regexp)
```

### Quelltext 7: Fehlermeldung aufgrund fehlgeschlagener CSRF-Prüfung

CSRF (**C**ross-**S**ite-**R**quest-**F**orgery) ist ein Angriff, bei dem der Angreifer den eingeloggten Zustand eines Nutzers in einer Anwendung ausnutzt. Um einen solchen Angriff festzustellen, sind in Alfresco einige Regeln definiert, die prüfen, ob der Nutzer auch von einer zugelassenen Seite kommt und nicht etwa durch eine Phishing-Attacke umgeleitet wurde.

Um die Prüfung anzupassen, muss zunächst ein Teil des Inhalts der Datei `share-security-config.xml`<sup>39</sup> in die Datei `share-config-custom.xml`<sup>40</sup> kopiert werden. Dies betrifft den kompletten Knoten `config`.

<sup>39</sup>/usr/local/tomcat/webapps/share/WEB-INF/classes/alfresco/share-security-config.xml

<sup>40</sup>/usr/local/tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml

```
1 <config evaluator="string-compare" condition="CSRFPolicy" replace="false
  ">
```

Quelltext 8: Beginn des CSRF-Knotens in der Konfigurationsdatei share-config-custom.xml

Dabei müssen noch folgende Änderungen vorgenommen werden. Unterhalb des `config`-Knotens befinden sich mehrere `rule`-Knoten, in denen für verschiedene URIs erlaubte *referer* und *origins* eingetragen sind. Die Platzhalter `{referer}` und `{origin}` werden durch die Adresse, unter der der Share erreichbar sein soll, bzw. den vollqualifizierten Domänennamen inklusive Protokoll, ersetzt. Das heißt konkret, dass, wenn Alfresco unter der Domäne *example.com* erreichbar sein soll, der Referer *https://example.com/\** und die Origin *https://example.com* ist.

```
1 <rule >
2   <request >
3     <method>POST</method>
4     <path >/page / dologin / </path >
5   </request >
6   <action name="assertReferer">
7     <param name="referer">https://example.com/*</param>
8   </action >
9   <action name="assertOrigin">
10    <param name="origin">https://example.com</param>
11  </action >
12 </rule >
```

Quelltext 9: Alfresco Share Konfiguration für share-config-custom.xml

Die in Quelltext 9 dargestellte Konfiguration ist nur ein Beispiel, die Werte müssen für alle Regeln angepasst werden. Änderungen können am besten über ein Dockerfile in das Image eingepflegt werden. Dazu muss zunächst die originale Datei aus einem laufenden Container

kopiert werden.

```
1 mkdir share /
2 docker ps | grep share // es wird die ID der Containers share angezeigt.
3 docker cp [id]:/usr/local/tomcat/shared/classes/alfresco/web-extension/
  share-config-custom.xml ./share/share-config-custom.xml
```

Quelltext 10: Kopieren der originalen Konfigurationsdateien aus einem laufenden Container

Anschließend wird der Inhalt der anzupassenden Konfigurationsdatei ausgegeben:

```
1 docker-compose exec share cat /usr/local/tomcat/webapps/share/WEB-INF/
  classes/alfresco/share-security-config.xml
```

Quelltext 11: Ausgeben des Inhalts der share-security-config.xml

Danach muss die Datei wie oben beschrieben angepasst und die Dockerfile erstellt werden:

```
1 FROM alfresco/alfresco-share:6.1.0
2
3 ADD ./share-config-custom.xml /usr/local/tomcat/shared/classes/alfresco/
  web-extension/share-config-custom.xml
```

Quelltext 12: Dockerfile für Alfresco Share

Mit diesem modifiziertem Image kann Alfresco nun über einen verschlüsselten und damit sicheren Kanal erreicht werden. <sup>41</sup>

### 2.1.2 Persistierung der Daten

Einer der Vorteile von Docker-Containern ist der flüchtige Speicher. Nach einem Neustart setzen sich die Container in den Ursprungszustand zurück. Dies gilt für alle Daten in dem

---

<sup>41</sup>Quelle: vgl. [Alfd]

Container, respektive auch für die Nutzerdaten. Damit ein Betrieb ohne Datenverlust möglich ist, müssen die Daten in Volumes ausgelagert werden.

Die Container *alfresco* und *solr6* können direkt angebunden werden. Die PostgreSQL-Datenbank versucht jedoch bei jedem Start eine neue Datenbank anzulegen, was fehlschlägt, wenn sich bereits Daten in dem Volume befinden. Deswegen muss der Datenbanksoftware mitgeteilt werden, dass eine bereits vorhandene Datenbank genutzt werden soll. Dies geschieht durch die Variable `PGDATA`. Ein Auszug der Konfiguration ist in Quelltext 13 zu sehen.

```
1  alfresco :
2      volumes :
3      - ./data/alf-repo-data : /usr/local/tomcat/alf_data /
4  postgres :
5      environment :
6      - PGDATA=/var/lib/postgresql/data/pgdata
7      volumes :
8      - ./data/postgres-data : /var/lib/postgresql/data /
9  solr6 :
10     volumes :
11     - ./data/solr-data : /opt/alfresco-search-services/data /
```

Quelltext 13: Auszug aus docker-compose.yml mit Konfiguration der Volumes

Bei initialem Start schreiben die Container ihre Daten in die Volumes. Werden die Container neu gestartet, bleiben die Daten erhalten.

Home Meine Dateien Freigegebene Dateien Sites Aufgaben Mitarbeiter Repository Admin-Tools Administrator Datelen, Personen, Sites such

### Administrator-Dashboard

#### Meine Sites

Alle Site erstellen

**Schneller Zugriff auf Ihre Sites**  
Eine Site ist ein Projektbereich, in dem Sie Inhalte für andere Mitglieder freigeben und besprechen können.  
Wir haben ein Beispiel für Sie aufgenommen. Damit können Sie die Funktionen in Alfresco Content Services erkunden.

**Sample: Web Site Design Project**  
This is a Sample Alfresco Team site.  
★ Favorit

#### Meine Aktivitäten

Alle Aktivitäten Alle Elemente in den letzten 7 Tagen

**Folgen Sie, was sich auf Ihren Sites tut**  
Durch die Aktivitäten in dieser Liste können Sie die Änderungen in den Sites, zu denen Sie gehören, verfolgen. Hier werden inhaltliche Aktualisierungen u. Ä. angezeigt. Sie können auch sehen, wer Ihren Sites betritt oder sie verlässt.

#### Meine Aufgaben

Aktive Aufgaben Workflow starten

<< < 1 - 3 von 3 > >>  
Aktive Aufgaben | Abgeschlossene Aufgaben

- Verify Adhoc Task Completed.  
Aufgabe abgeschlossen, noch nicht gestartet
- Verify Adhoc Task Completed.  
Aufgabe abgeschlossen, noch nicht gestartet
- Adhoc Task  
Aufgabe, noch nicht gestartet

#### Meine Dokumente

Habe ich kürzlich geändert

**Verfolgen Sie Ihre eigenen Inhalte**  
Dieses Dashlet listet unabhängig von der Site Inhalte auf, die Ihnen wichtig sind. Verwenden Sie die Filter, damit Sie leicht finden, was Sie suchen.

Abbildung 6: Startseite *Alfresco Share*

## 2.2 Addons

Ein Addon ist eine Zusatzkomponente. Sie „erweitert [...] Software mit optionalen Modulen, wobei nur die vorhandenen Bibliotheken der jeweiligen Anwendung genutzt und um bestimmte Funktionen bereichert werden.“<sup>42</sup>

Alfresco lässt sich durch eine Vielzahl an Addons erweitern. Diese sind in einem Online-Store abrufbar und zumeist kostenfrei. Es sollte bei jedem Addon geprüft werden, ob es für einen Einsatz beim Kunden geeignet und ob dies durch die Lizenzbestimmungen erlaubt ist. Außerdem birgt die Installation von Addons das Risiko, dass - beabsichtigt oder nicht - schadhafter Code im Unternehmen ausgeführt wird. Es ist nicht empfohlen, sich darauf zu verlassen, dass ein Addon fehlerfrei funktioniert, nur weil es eine Vielzahl an Installationen hat. Ein Fall im September 2017 bei einem Plugin für die Blogsoftware *WordPress* zeigt, dass auch ein vertrauenswürdigen Plugin zu einem späteren Zeitpunkt einen anderen Sicherheitsstatus erlangen kann. Im o.g. Fall wechselte der Besitzer des Plugins *Display Widgets*, welcher daraufhin Schadcode in das Plugin einschleuste und damit administrativen Zugriff auf über 200.000 Webseiten erlangte.<sup>43</sup>

Alle verfügbaren Addons von Alfresco können auf der offiziellen Community-Seite<sup>44</sup> eingesehen werden. Allerdings muss sich ein Betreiber der Software bewusst sein, dass diese zentrale Anlaufstelle eine enorme Abhängigkeit darstellt. Im Mai 2018 wurde die Vorgängerseite ohne Vorankündigung offline geschaltet. Begründet wurde dies mit der Datenschutzgrundverordnung. Erst über eine Woche später wurden die Plugins in einem Communitybereich wieder veröffentlicht.<sup>45</sup>

Jedes Addon unterliegt seinem eigenen Updatezyklus und muss händisch aktualisiert werden. Zudem muss bei jedem einzelnen Addon darauf geachtet werden, dass Aktualisierungen rechtzeitig bekannt werden. Dies ist vor allem bei kritischen Sicherheitslücken wichtig. Mit

---

<sup>42</sup>Quelle: [Sof]

<sup>43</sup>Quelle: vgl. [pre19]

<sup>44</sup>Online verfügbar unter: <https://community.alfresco.com/community/alfresco-addons/content>

<sup>45</sup>Quelle: vgl. [ino18]

jedem Update der Kernsoftware Alfresco besteht die Gefahr, dass Addons, die zuvor funktionierten, nicht mehr kompatibel sind. Dies geschieht häufig bei Addons, die lange nicht mehr vom Entwickler aktualisiert worden sind. Diese nicht mehr aktiv entwickelten Addons sollten auf keinen Fall mehr betrieben werden, da sogar bekannte Sicherheitslücken nicht mehr geschlossen werden. Selbst wenn das Addon funktional unentbehrlich ist und es keine Alternative dazu gibt, muss sichergestellt werden, dass kein potentiell unsicherer Code in Betrieb ist. Um zu vermeiden, dass nicht mehr aktualisierte Addons betrieben werden, sollte man bereits bei der Auswahl beachten, dass Addons in Zukunft abgelöst werden müssen. Um zügig einen Technologiewechsel vollziehen zu können, sollte bei der Verwendung eines Addons immer eine Alternative bekannt sein. Außerdem muss immer die Frage gestellt werden, ob und wie Support zu diesem Addon möglich ist. Dies umfasst nicht nur die Installation, sondern auch den Betrieb und vor allem das Beheben von Störungen. Mittels eines Supportvertrages stellt der Auftraggeber des benannten Projektes sicher, dass eventuell entstehender Schaden durch Anbieter des Addons übernommen wird.

Abschließend lässt sich sagen, dass Addons zwar viele Funktionen hinzufügen, welche die Kernsoftware nicht besitzt. Alfresco kann genau an die Bedürfnisse und den Einsatzzweck angepasst werden. Es gibt aber auch viele Risiken und Gefahren, denen man sich jederzeit bewusst sein muss. Dadurch entsteht ein hoher Aufwand in Analyse, Bewertung, Betrieb und Wartung eines Addons. Als Alternative besteht immer die Möglichkeit, ein Addon selbst zu entwickeln. Dadurch steigt zwar der initiale Aufwand bei der Einführung des Addons, Wartungen vereinfachen sich dadurch jedoch und es entstehen keine neuen Abhängigkeiten von Dritten. Es ist also genau zu evaluieren, ob ein Addon eingesetzt wird und ob auf eine fertige Lösung oder auf eine Eigenentwicklung zurückgegriffen werden soll.

Dem Kunden wurden die o.g. Aspekte kommuniziert und gemeinsam die Entscheidung getroffen, dass fehlende Funktionen durch Eigenentwicklungen bereitgestellt werden sollen. Ausschlaggebend waren die überwiegend negativen Aspekte von Addons und die damit einhergehenden Gefahren.

## 2.3 Content Repository

Alfresco speichert und verwaltet Dokumente im sog. *Content Repository*. Folgende Dienste werden vom Repository bereitgestellt:

- Erzeugung, Modifikation, Löschen von Inhalten, Metadaten und deren Beziehungen untereinander,
- Abfrage von Content,
- Zugriffskontrolle auf Inhalte,
- Versionierung und Sperren von Inhalten und
- Import/Export von Dokumenten.<sup>46</sup>

Die gespeicherten Inhalte, Metadaten und Beziehungen werden in einer *PostgreSQL* Datenbank gespeichert.<sup>47</sup>

## 2.4 Interaktion mit anderen Systemen

Das DMS soll nicht nur Dokumente verwalten, welche in das System geladen werden, sondern auch mit bestehenden Systemen interagieren.

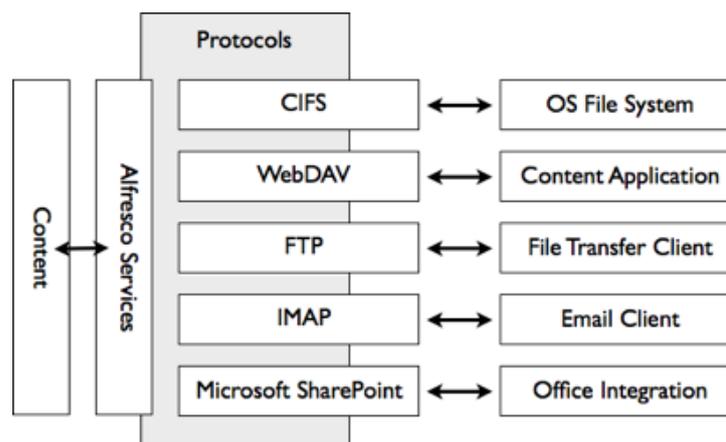


Abbildung 7: Zusammenspiel von Content, Protokollen und Clients<sup>48</sup>

<sup>46</sup>Quelle: [Hol15], Seite 15 (gekürzt)

<sup>47</sup>Quelle: vgl. [Hol15], Seite 16

Um eine große Bandbreite an Nutzungsmöglichkeiten zu bieten, besitzt *Alfresco* eine Vielzahl an Schnittstellen. Über diese können Inhalte abgerufen, geschrieben oder geändert werden. Außerdem ist es über die meisten Schnittstellen möglich, die Ordnerstruktur zu ändern, vorherige Versionen abzurufen oder eine Suche auszuführen.<sup>49</sup> Die Schnittstellen, die *Alfresco* konkret unterstützt, sind in der Abbildung 7 dargestellt.

### 2.4.1 Anbindung an KeyCloak

Der Kunde verwaltet seine Nutzeraccounts in einer freien Nutzerverwaltungssoftware. *KeyCloak* stellt eine Alternativmöglichkeit zum *Microsoft Windows Active Directory* dar.<sup>50</sup> Es bietet eine zentrale Verwaltung von Nutzerdaten wie Name, E-Mail-Adresse und Passwort und wickelt den Login für andere Anwendungen ab. Der Kunde möchte auch die im *Alfresco* hinterlegten Nutzer über *KeyCloak* verwalten, da bereits Erfahrung im Umgang mit diesem System ist. Außerdem reduziert sich der Pflegeaufwand, da alle Nutzer des Kunden in nur einer Software verwaltet werden. *KeyCloak* bietet zudem eine Vielzahl an Sicherheitsmechanismen. Das ist insbesondere wichtig, da sich externe Dienstleister des Kunden registrieren können sollen und somit die Software frei im Internet zugänglich ist. Eine eigene Implementierung entfällt somit.

Zunächst muss im *KeyCloak* ein sog. *Realm* angelegt werden. Das ist vergleichbar mit einer Domäne in einem *Windows*-Netzwerk. Unter diesem *Realm* werden Nutzer, Gruppen und Anwendungen verwaltet. Da *KeyCloak* für *Alfresco* speziell konfiguriert werden muss, empfiehlt es sich, den von *Alfresco* bereitgestellten *Realm* in *KeyCloak* zu importieren.<sup>51</sup>

Der angelegte *Realm* ist sofort einsatzbereit. Im Reiter *Users* sind bereits Beispielnutzer hinterlegt. Es können hier weitere Benutzer angelegt und Passwörter geändert werden.

Jede Anwendung, die mit dem *KeyCloak* kommuniziert, wird als *Client* bezeichnet. Im Rei-

---

<sup>48</sup>Quelle: [Alff]

<sup>49</sup>Quelle: vgl. [Hol15], Seite 16

<sup>50</sup>Quelle: vgl. [Key]

<sup>51</sup>online verfügbar als JSON-Datei unter <https://github.com/Alfresco/alfresco-identity-service/blob/master/helm/alfresco-identity-service/alfresco-realm.json>

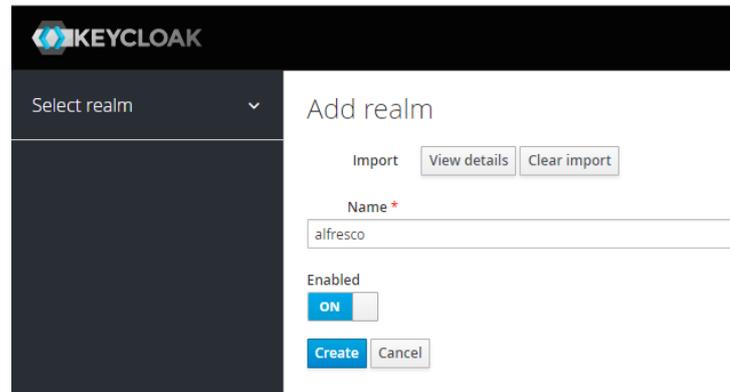


Abbildung 8: Alfresco Realm in KeyCloak importieren

ter *Client* ist bereits der Client *activiti* definiert, der für die Anmeldung aller Alfresco-Module genutzt werden kann.

Alfresco wird über die *docker-compose.yml* so konfiguriert, dass KeyCloak als Authentifizierungsservice vor den Login geschaltet wird. Für den Endnutzer ist dies nicht erkennbar, da weiterhin die Loginmasken der jeweiligen Alfresco-Dienste genutzt werden. Lediglich die Validierung der Logindaten wird geändert und auf KeyCloak umgelenkt. Dazu wird die Variable `JAVA_OPTS`, wie in Quelltext 14 ersichtlich, erweitert. Nachdem der Login von KeyCloak freigegeben wurde, wird der Nutzer in den gesicherten Bereich nach dem Login weitergeleitet.

```
1  environment :
2    JAVA_OPTS : "
3      ...
4      -Dauthentication.chain=identity-service1:identity-service,
5          alfrescoNtlm1:alfrescoNtlm
6      -Didentity-service.enable-basic-auth=true
7      -Didentity-service.authentication.validation.failure.silent=
8          false
9      -Didentity-service.auth-server-url=https://example.com/auth
10     -Didentity-service.realm=alfresco
11     -Didentity-service.resource=activiti
12     -Didentity-service.public-client=true
```

```

11     -Didentity-service.ssl-required=none
12     "

```

#### Quelltext 14: Erweiterung der Umgebungsvariablen des Containers alfresco

Die Werte für *auth-server-url*, *resource* („Client“ im KeyCloak) und *realm* müssen ggf. angepasst werden.

Nach einem Neustart aller Container kommuniziert Alfresco nun mit dem KeyCloak. Alle in Alfresco angelegten Nutzer funktionieren weiterhin. Existiert ein Nutzernamen sowohl in Alfresco als auch im KeyCloak, so funktionieren beide Passwörter, es handelt sich technisch **nicht** um zwei verschiedene Nutzer. Deshalb sollte die Nutzerverwaltung des Alfresco initial geleert werden, um eine bessere Kontrolle über die Nutzerverwaltung zu haben. Werden Nutzer im KeyCloak angelegt, so erscheinen diese nach dem ersten Login ebenso in der Nutzerverwaltung von Alfresco.

Abbildung 9 zeigt zwei Screenshots von Passwort-Änderungsmasken. Teil (a) zeigt die 'Passwort ändern' Maske für Alfresco-Nutzer mit den Feldern 'Altes Passwort eingeben', 'Neues Passwort eingeben' und 'Neues Passwort bestätigen', sowie 'OK' und 'Abbrechen' Buttons. Teil (b) zeigt die 'Foto' Maske für Keycloak-Nutzer mit den Feldern 'Kontingent', 'Neues Passwort', 'Passwort prüfen' und einem 'Konto deaktivieren' Checkbox.

Abbildung 9: Maske zum Passwort ändern für Alfresco- (a) und Keycloak-Nutzer (b)

Wird ein Passwort eines Nutzers in Alfresco geändert, so bezieht sich diese Änderung immer nur auf den Alfresco-Nutzer. Passwörter von Nutzern, die im KeyCloak verwaltet werden, können nicht in Alfresco geändert werden. Eine Nutzersession, die in einem Alfrescomodul geöffnet wurde, wird nicht auf andere Module übertragen. Das KeyCloak übernimmt kein Sessionhandling, sondern führt nur den Login-Prüfprozess aus. Für die Umsetzung im Kun-

denprojekt bedeutet das, dass zukünftige zusätzlich genutzte Module den Loginvorgang über die gleiche Maske abwickeln müssen wie der Share.

### 2.4.2 Anbindung an inubit

Der Kunde nutzt zur Prozessierung von Dokumenten derzeit die Software *inubit*. Die sog. *BPM Suite* kann eine Vielzahl an Aufgaben übernehmen, darunter auch die Freigabe von Dateien. Dieser Prozess wird in *inubit* mit einer eigenen Webseite pro Freigabeprozess umgesetzt und soll an Alfresco ausgelagert werden, da er sich dadurch stark vereinfacht und somit Entwicklungskosten und Aufwand eingespart werden.

Um die beiden Systeme miteinander zu verknüpfen ist es möglich, eine gemeinsame Dateiablage einzurichten. Dabei kann es sich um einen Netzwerkspeicher handeln, über den die Kommunikation stattfindet. Es ist auch möglich, die Dateien über APIs der beiden Programme auszutauschen.

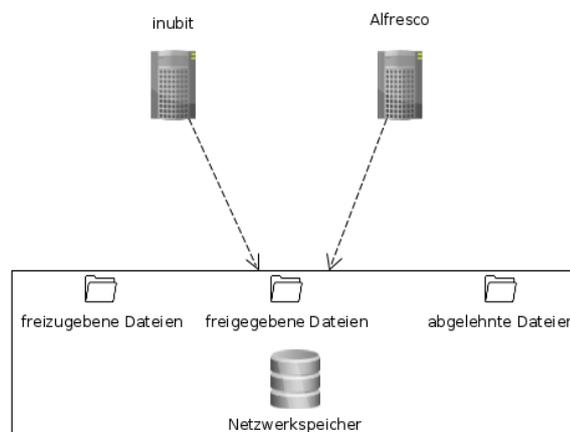


Abbildung 10: Systemdiagramm: Zusammenspiel Alfresco und inubit

Im Folgenden wird erläutert, wie Alfresco konfiguriert wird, damit es Freigabeprozesse aus dem *inubit* über eine Dateiablage übernimmt.

In der Abarbeitung stößt *inubit* zunächst eine beliebige Prozessierung an, in deren Verlauf eigentlich ein Freigabeprozess ausgeführt werden würde. Dieser Teilprozess wird dahingehend

ersetzt, dass inubit stattdessen über einen *VFS-Connector*<sup>52</sup> die freizugebende Datei auf einen Netzwerkspeicher schreibt, der von Alfresco überwacht wird. Die Software inubit entscheidet selber nicht, ob die Datei diesen Freigabeprozess überhaupt durchlaufen muss. Diese Entscheidung wird in Alfresco anhand von vorher definierten Regeln getroffen. Aus diesem Grund wird die Datei in jedem Fall auf den Speicher geschrieben. Die Ablage der Datei triggert dann eine Regel, die zunächst überprüft, ob eine Freigabe notwendig ist. Dies kann anhand von beliebigen Kriterien, wie beispielsweise Dateieigenschaften (Name, Größe, Anzahl Seiten, Inhalt, ...) festgemacht werden. Ist ein Freigabeprozess notwendig, so wird ein sog. Workflow in Alfresco gestartet, der dem jeweiligen Bearbeiter zugewiesen wird. Dieser beinhaltet die freizugebende Datei.

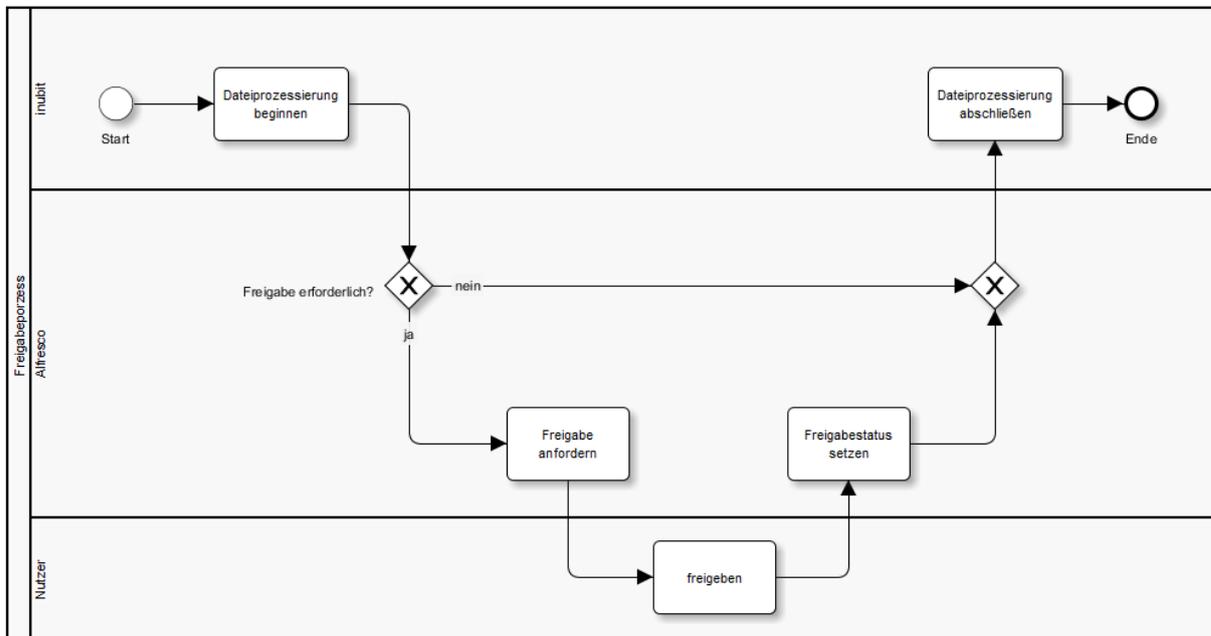


Abbildung 11: Freigabeprozess inubit und Alfresco (fachlich)

Ist die Freigabe in Alfresco erfolgt, wird die Datei in einen anderen Ordner verschoben. Wurde die Freigabe abgelehnt, so wird die Datei in einen dritten Ordner verschoben. In beiden Fällen bleibt die originale Datei nicht erhalten. Um den Prozess im inubit fortzusetzen, läuft dort ein sog. *Scheduler*, der periodisch die beschriebenen Ordner abrufen und die Dateiprozessierung fortsetzt. Der 0Prozess ist in Abbildung 12 fachlich und in Anhang A technisch beschrieben.

<sup>52</sup>Begriffserklärung s. Glossar auf Seite 41

## 2.5 Corporate Design

Für eine bessere Integration von Alfresco in die vorhandene Systemlandschaft solle das Design der Software an das *Corporate Design*<sup>53</sup> des Kunden angepasst werden. Ein „Corporate Design [...] bezeichnet ein konzeptioniertes, einheitliches Erscheinungsbild eines Unternehmens oder einer Marke. Dazu gehören grafisch abgestimmte Produktverpackungen und Geschäftsausstattung, aber auch optisch passende Marketingmaßnahmen.“<sup>54</sup> Der Kunde wünscht eine Designanpassung, um ein geschlossenes Bild von seiner Firma darzustellen. Auftragnehmer des Kunden kennen bereits vorhandene Oberflächen und können sich somit in dazukommenden ebenfalls zurecht finden.



Abbildung 12: Styleauswahl in Alfresco

Um das Design der für den Kunden sichtbaren Bereich (Share) anzupassen, wird zunächst eines der in Alfresco standardmäßig vorhandenen Designs kopiert und entsprechend an das Corporate Design des Kunden angeglichen. Eine detaillierte Anleitung dazu wird auf den Dokumentationsseiten von Alfresco<sup>55</sup> angeboten.

Nach Ausführung der in diesem Kapitel beschriebenen Schritte ist die Installation von Alfresco abgeschlossen. Allen Kundenvorgaben konnte entsprochen werden. Im Folgenden wird nun auf die Umsetzung der inhaltlichen Kundenforderungen eingegangen.

<sup>53</sup>Begriffserklärung s. Glossar auf Seite 41

<sup>54</sup>Quelle: [Die11]

<sup>55</sup>Online verfügbar unter: <https://docs.alfresco.com/5.1/tasks/dev-extensions-share-tutorials-add-theme.html>

## 3 Fachliche Umsetzung

### 3.1 Regeln

Regeln sind automatische Aktionen, die von Alfresco ausgeführt werden, sobald ein Trigger eintritt. Ein Trigger ist dabei immer ein Ereignis, wie beispielsweise die Ablage oder das Aktualisieren einer Datei in einem Ordner. Zusätzlich können zu jedem Trigger auch Bedingungen definiert werden, um beispielsweise zu lange Dateinamen zu kürzen. Je nach definierter Regel wird Alfresco dann eine Aktion ausführen. Dies kann das Verschieben, Umbenennen oder Konvertieren des Dokuments sein, das Starten eines Workflows oder setzen von Tags oder Eigenschaften.

Neue Regel \* Pflichtfelder

**Allgemein**

Name: \*

Beschreibung:

**Regel definieren**

Wenn:

Wenn alle Kriterien erfüllt sind:

Wenn nicht alle Kriterien erfüllt sind:

**Aktion ausführen:**

**Weitere Optionen**

Regel abschalten  
 Regel im Hintergrund ausführen  
 Regel trifft auf Unterordner zu  
Falls ein Fehler auftritt, Script ausführen:

Abbildung 13: Maske zur Erstellung einer Regel in Alfresco

Regeln können für Ordner oder einzelne Dateien angelegt werden. Damit sind Prozesse in Alfresco hochskalierbar und können an nahezu alle Anwendungsfälle angepasst werden.

In Abbildung 13 wird die Maske zur Erstellung einer Regel gezeigt. Einzelne Bedingungen und Aktionen lassen sich über den grafischen Editor einfügen.

Wie bereits in der Einleitung dieser Arbeit erwähnt, kann die elektronische Speicherung von Dokumenten viele Vorteile gegenüber der analogen Datenhaltung bringen. Jedoch müssen zunächst alle Papierdokumente digitalisiert, in Alfresco importiert und sortiert werden.

Alfresco vereinfacht diesen Prozess maßgeblich, in dem es Funktionen bereitstellt, die kombiniert ohne großen Aufwand Papierdokumente in das System übertragen können. Es genügt zunächst die Dokumente (automatisiert) zu scannen und als PDF-Dokument in einem Netzwerkspeicher abzulegen. Dieser Speicher kann in Alfresco eingebunden, gelesen und die daran enthaltenen Dokumente prozessiert werden. Durch die in Alfresco enthaltene OCR<sup>56</sup> können diese Dokumente durchsuchbar gemacht und deren Inhalte erkannt werden. Durch vorher definierte Regeln können diese Dokumente nun verschoben, geordnet und ggf. einer manuellen Sichtprüfung zugewiesen werden. Um den Vorgang anzustoßen, müssen lediglich Regeln in Alfresco angelegt und die Dokumente durch den Scanner in das überwachte Verzeichnis abgelegt werden.

### 3.2 Zugriffskontrolle Dokumente

In Alfresco Share existieren sog. *Sites*. Diese sind mit einem Projektraum in Microsoft Sharepoint zu vergleichen. Das bedeutet, dass voneinander unabhängige Projekte getrennt existieren. Nutzer müssen explizit Zugriffsrechte auf einzelne Sites erhalten, um Dateien betrachten oder verändern zu können. Weiterhin können im gesamten Repository Rechte nutzer- oder gruppenbezogen gesetzt werden. Dies kann sowohl auf Projekten und Ordnern, als auch auf einzelnen Dateien geschehen. Damit kann also Datenschutz konform pro Datei explizit eingestellt werden, welcher Nutzer welche Aktion mit welcher Datei machen darf.

Dazu gibt es in Alfresco verschiedene Rollen, in denen definiert ist, ob ein Nutzer eine Datei nur lesen oder auch beschreiben darf. Diese Freigabe bezieht sich sowohl auf nutzereigene

---

<sup>56</sup>optische Texterkennung, Begriffserklärung s. Glossar auf Seite 41

Dateien, als auch auf Dateien anderer Nutzer. Diese Rollenverteilung bildet die Grundlage für die Rechtekonfiguration auf allen Ebenen.<sup>57</sup>

Daten dürfen immer nur für berechtigte Personen zugänglich sein. Vielmehr muss sich nun die Frage gestellt werden, was passiert, wenn alle Mitarbeiter mit Zugriffsrechten auf eine Datei oder einen Ordner arbeitsunfähig werden. In diesem Fall hat niemand anderes Zugriff auf den Account des betroffenen Mitarbeiters. Laut *Forschungsstelle Recht im DFN*<sup>58</sup> darf ein Softwareaccount nicht ohne Weiteres übernommen werden, da diese Handlung ggf. gegen das Fernmeldegeheimnis verstößt. Der Kunde möchte eine Möglichkeit haben, auf legale Art und Weise alle Daten im Notfall erreichen zu können. Da aber auf spezifische Dateien nur von diesem Account aus zugegriffen werden darf, muss eine Möglichkeit geschaffen werden, die Zugriffsrechte im Notfall ändern zu können.

In Alfresco darf nicht einfach das Passwort zu einem Mitarbeiteraccount zurückgesetzt werden darf, sondern ein anderer Weg muss gefunden werden, auf Daten zuzugreifen. Es besteht zwar immer die Möglichkeit über das Dateisystem die Dateien abzurufen, dies ist aber eine technisch unsaubere Lösung, da der Dateistatus bei einer Änderung in Alfresco nicht angepasst wird. Es kann also zu Unstimmigkeiten in der Datenhaltung kommen, was zu einem unerwartetem Verhalten oder einem Datenverlust führen kann. Deswegen bietet Alfresco die Möglichkeit, einen sog. *Super User* anzulegen. Dabei handelt es sich um Accounts mit speziellen Rechten. Es können Nutzer auf Sites berechtigt werden, ohne dass es der Zustimmung des jeweiligen Sitebesitzers bedarf. Sich im Repository der Site befindliche Dateien in Unterordnern werden automatisch mitberechtigt und die darunterliegenden Berechtigungsstufen damit übergangen. Somit kann auf jede Datei im Repository der Seite ein Zugriff durch einen Dritten gewährleistet werden.<sup>59</sup> Dieser Sonderzugriff ist jedoch nur dann Datenschutz konform, wenn es sich um einen Notfall handelt und dieser zwingend notwendig ist.<sup>60</sup>

---

<sup>57</sup>Quelle: [Alfc]

<sup>58</sup>Quelle: vgl. [For], Seite 9

<sup>59</sup>Quelle: vgl. [Alfb]

<sup>60</sup>Quelle: vgl. [For], Seite 9

### 3.3 Sicherung der Alfresco-Daten

Grundsätzlich werden Dateien, Ordner und andere Inhalte, die in Alfresco gelöscht werden, in einen Papierkorb verschoben. Damit besteht zunächst die Möglichkeit, die Daten wiederherzustellen. Damit eine Datei endgültig aus dem System gelöscht wird, muss sie aus dem Papierkorb entfernt werden. Um zu verhindern, dass Dateien vorschnell gelöscht werden, befindet sich der Papierkorb nicht auf der Seite einer Dateiablage, sondern auf der Seite der Profileinstellungen. Es ist damit nicht möglich, versehentlich Daten endgültig zu löschen.

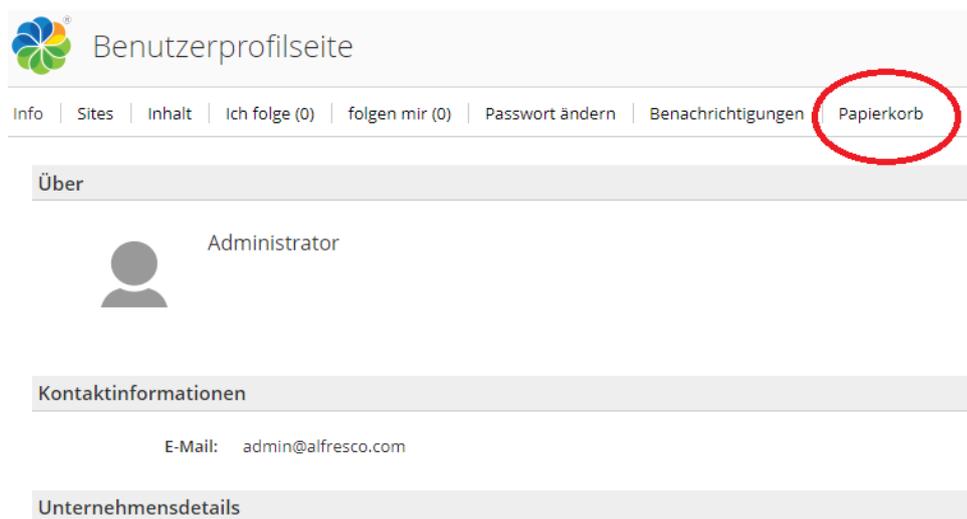


Abbildung 14: Der Papierkorb in Alfresco befindet sich auf der Profileseite.

Der Papierkorb ist jedoch nur eine Zwischenstufe im Prozess des Löschen von Dateien und keine Sicherungskopie. Ein Backup erfolgt auf einem externen Datenträger, nicht auf dem, der die Quelldaten speichert. Je nach Sicherheitsanforderung müssen die Datenträger physisch eine gewisse Entfernung haben, damit beispielsweise ein Brand in einem Rechenzentrum nicht den gesamten Datenbestand unwiederbringlich vernichtet.

Exemplarisch werden zwei Möglichkeiten der Datensicherung gezeigt, die für den Kunden relevant sind.

### 3.3.1 Replikation

Alfresco bietet die Möglichkeit, das gesamte Repository über sog. *Replikationsjobs* zu klonen. Dabei werden für den jeweiligen Job ausgewählte Daten an einen Zielort kopiert. Dies ist vergleichbar mit dem Synchronisieren eines Ordners auf ein externes Laufwerk. Bei der Konfiguration des Replikationsjobs kann ausgewählt werden, wann der Job ausgeführt und wann, bzw. wie oft er wiederholt werden soll.

Neuen Replikationsjob anlegen \* Pflichtfelder

**Allgemein**

Name: \*

Beschreibung:

**Nutzdaten**

Quelldokumente:

**Übertragungsziel**

Übertragen nach:

**Ablaufplan**

Job planen

**Weitere Optionen**

Aktiviert

Abbildung 15: Maske zum Anlegen eines Replikationsjobs

Die Aufgaben werden durch Cronjobs<sup>61</sup> ohne aktives Handeln eines Nutzers ausgeführt. Der Vorteil dieser Methode liegt in der sehr einfachen Konfiguration. Nachteilig ist, dass ausschließlich Repositorydaten kopiert werden. Es findet keine Kopie der Systemdaten, wie beispielsweise Sites oder Nutzerberechtigungen statt.

<sup>61</sup>Begriffserklärung s. Glossar auf Seite 41

### 3.3.2 Backup

Alfresco speichert Repository-Daten auf dem Dateisystem im Ordner *alf\_data*. Daten, wie Nutzerrechte, Sites und Regeln werden in einer Datenbank gespeichert. Um eine unabhängige Sicherungskopie der Daten zu erstellen, müssen diese aus dem Ordner *alf\_data* kopiert werden.<sup>62</sup> Außerdem muss die zugehörige Datenbank, in der Metadaten, wie Nutzerberechtigungen, gespeichert sind, exportiert werden. Diese Daten werden von der *Speicherengine* der Datenbank verwaltet und im Standardordner persistiert. In einer mit Docker umgesetzten Landschaft liegen alle Daten in den Volumes der jeweiligen Container auf dem Dateisystem. Da sich dort alle Daten befinden, genügt es, sie in einem regelmäßigen Abstand auf einen unabhängigen Datenträger zu kopieren, von dem aus die Daten im Wiederherstellungsfall gelesen werden können.<sup>63</sup>

Die Vorteile der unabhängigen Sicherungsmethode liegen darin, dass die Daten in frei gewählten Backup-Strategien gesichert werden können. Es gibt keinerlei Einschränkungen bezüglich inkrementeller, differenzieller oder Vollbackups. Außerdem können die Ziellandschaft und das Übertragungsprotokoll frei gewählt werden. Es besteht eine absolute Unabhängigkeit zu Alfresco und dessen bereitgestellten Methoden und ist deshalb zu empfehlen.

---

<sup>62</sup>Quelle: [Hol15]

<sup>63</sup>Quelle: [Nor17]

## 4 Zusammenfassung

Ziel dieser Arbeit war es, im Kundenauftrag einen Plan zu erstellen, wie ein Dokumentenmanagementsystem installiert und angepasst werden kann. Dabei zu beachten waren die Kundenanforderungen an das Dokumentenmanagement. Im Wesentlichen sollte untersucht werden, ob Alfresco für den Kunden eingesetzt werden kann, welchen Anforderungen die Software entspricht und welche Funktionen nachgerüstet werden müssten. Dazu wurden eingangs die Kundenanforderungen definiert. Diese waren direkt vom Kunden vorgegeben und nicht optional.

Zunächst musste untersucht werden, ob ein Einsatz eines DMS überhaupt notwendig ist. Es wurde anhand von wissenschaftlichen Quellen aufgezeigt, welcher Nutzen mindestens erfüllt sein muss, damit ein Betrieb wirtschaftlich ist. Die Untersuchung zeigte, dass ein DMS ein hohes Einsparpotential an Zeit und Kosten beim Austausch von Dokumenten ermöglicht. Aufgrund der zu erwarteten Nutzerzahl ist eine digitale Lösung für den Kunden faktisch alternativlos.

Im weiteren Verlauf der Untersuchungen wurde festgestellt, dass die kostenfreie Version von Alfresco ausreichend für dieses Kundenprojekt ist. Mögliche Vorteile, wie zusätzliche Module, werden nicht benötigt und würden keinen reellen Mehrwert bieten.

Technische und fachliche Anforderungen wurden getrennt betrachtet, um eine genauere Untersuchung zu ermöglichen.

Im Rahmen der technischen Anforderungen wurde die Installation via Docker-Containern und die Einrichtung der Software über einen verschlüsselten Kommunikationskanal betrachtet. Um einen Datenverlust zu vermeiden, wurde erklärt, wie die Nutzdaten in Volumes persistiert und durch geeignete Backupmethoden gesichert werden.

Da Alfresco öffentlich im Internet erreichbar ist, muss es gegen Angriffe geschützt werden. Dazu wurde erklärt, wie CSRF-Prüfungen einzurichten sind.

Abschließend wurde in diesem Kapitel untersucht und erklärt, ob und wie Alfresco mit den

sich bereits in Betrieb befindlichen Softwares inubit und KeyCloak kommunizieren kann.

Im Anschluss wurde geprüft, ob Alfresco auch den fachlichen Anforderungen gerecht wird. Dabei wurde insbesondere auf die Punkte Zugriffskontrolle und Datensicherheit eingegangen. Es wurde beschrieben, wie mit automatisierten Regeln Dokumente digitalisiert werden können und wie durch eine Replikation eine Alfresco-Instanz kopiert werden kann. Durch diese Untersuchungen wurde festgestellt, dass Alfresco die geforderten Funktionen vollständig bereitstellt. Zukünftige, mit der Software realisierbare Funktionsanforderungen werden in Kundenabsprache als Eigenentwicklung umgesetzt.

Während der Einrichtung von Alfresco traten einige Probleme auf. Bereits während der Installation wurde festgestellt, dass Alfresco selbst in einer Dockerumgebung mit konfiguriert werden muss, bevor es überhaupt zu einem Login kommen kann. Dazu wurde Wissen benötigt, welches sich zunächst erst angeeignet werden musste. Dies führte vor allem dazu, dass zu Beginn des praktischen Teils zeitliche Verzögerungen eintraten, die später jedoch wieder aufgeholt werden konnten.

Weiterhin traten Probleme bei der Einrichtung der Kommunikation zwischen Alfresco und KeyCloak auf. Zwar wurden die Kundenvorgaben erreicht, jedoch konnte nicht das volle Potential der Nutzerverwaltungssoftware ausgeschöpft werden. Als Beispiel dafür ist das Session-Handling zu nennen, welches bereits in Kapitel 2.4.1 beschrieben wurde. Außerdem konnte keine vollständige Synchronisation der Nutzer eingerichtet werden. Änderungen in einem System werden nicht auf das jeweils andere übertragen. Eine Möglichkeit, die Synchronisation mit Einschränkungen dennoch zu ermöglichen wurden in eben diesem Kapitel beschrieben.

Als Ergebnis dieser Masterthesis wurde Alfresco als Möglichkeit für den Projekteinsatz bestätigt. Dem Kunden wurde ein Beispiel für eine `docker-compose.yml`<sup>64</sup>, sowie die dazugehörige Ordnerstruktur übergeben. Alle Ziele dieser Arbeit wurden erfüllt. Deswegen kann die Aufgabenstellung als erfolgreich abgeschlossen angesehen werden.

---

<sup>64</sup>s. Anhang B: Vollständiges docker-dompose File mit gesamter Alfresco-Umgebung auf Seite 44

## Glossar

- ajp:** Das **Apache JServ Protocol** ist ein Protokoll zur Kommunikation zwischen Web- und Applikationsservern.
- Clustering:** Der Begriff *Clustering* bezeichnet das Zusammenführen mehrerer Server zu einer Instanz, zur Ausführung einer Anwendung.
- Corporate Design:** Der Begriff *Corporate Design* bezeichnet ein standardisiertes Design, meist innerhalb einer Firma. Es definiert verwendete Formen und Farben, Positionen und Bilder. Diese wiederkehrenden Elemente sollen eine Wiedererkennung einfach machen.
- Cronjob:** Ein **Cronjob** ist eine meist periodisch wiederholte automatisch ausgeführte Aufgabe.
- JMX:** **Java Management Extensions** ist eine Spezifikation zur Verwaltung und Überwachung von Java-Anwendungen.
- OCR:** Eine **Optical Character Recognition** (deutsch: optische Texterkennung) ist eine Technik, um Texte aus einem Bild zu erkennen.
- OpenSource:** Eine Software wird als "OpenSource" bezeichnet, wenn ihr Quelltext öffentlich einsehbar ist, geändert und genutzt werden darf. Nicht jede OpenSource-Software ist auch kostenlos.

- PID:** Eine **Prozess ID** ist eine eindeutige Identifikationsnummer eines laufenden Prozesses während der Ausführungszeit eines Betriebssystems.
- Repository:** Ein "Repository" ist eine hierarchische Dateiablage.
- Spring Framework:** Das Spring Framework bietet ein umfassendes Programmier- und Konfigurationsmodell für moderne Java-basierte Unternehmensanwendungen.<sup>65</sup>
- URI:** Ein **Uniform Resource Identifier** ist die Angabe in einer URL hinter dem Host und gibt die Location an.
- VFS-Connector:** Der VFS-Connector (Virtual File System) ist ein Modul der inubit IS Suite und „sorgt dafür, dass die inubit Process Engine mit lokalen und entfernten Dateisystemen, wie z. B. Windows Shared Verzeichnissen oder dem lokalen Repository kommunizieren kann. Der Konnektor unterstützt dafür das CIFS/Samba- und das Repository-Protokoll.“<sup>66</sup>

---

<sup>65</sup>Quelle: vgl. [Piv19]

<sup>66</sup>Quelle: [Bos]

# Anhang

## Anhang A: Freigabeprozess inubit und Alfresco (technisch)

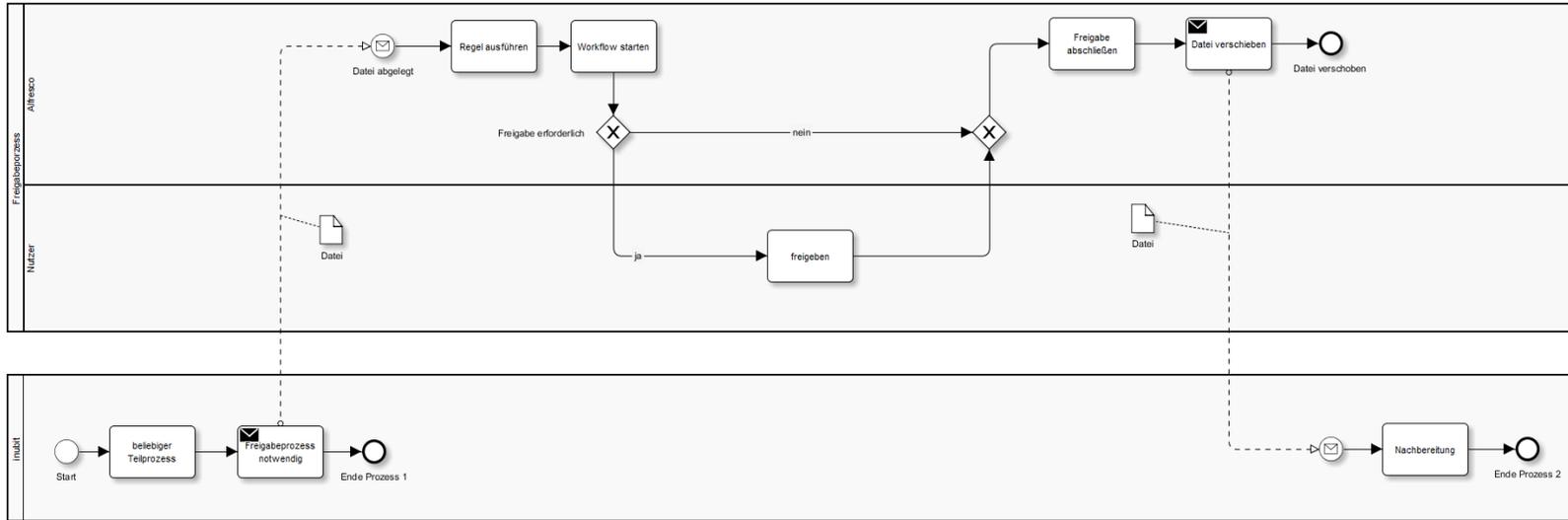


Abbildung 16: Freigabeprozess inubit und Alfresco (technisch)

## Anhang B: Vollständiges docker-compose File mit gesamter Alfresco-Umgebung

```
1  version: "3.7"
2
3  services:
4    alfresco:
5      image: alfresco/alfresco-content-repository-community:6.1.2-ga
6      #mem_limit: 1500m
7      environment:
8        JAVA_OPTS : "
9          -Ddb.driver=org.postgresql.Driver
10         -Ddb.username=alfresco
11         -Ddb.password=alfresco
12         -Ddb.url=jdbc:postgresql://postgres:5432/alfresco
13         -Dsolr.host=solr6
14         -Dsolr.port=8983
15         -Dsolr.secureComms=none
16         -Dsolr.base.url=/solr
17         -Dindex.subsystem.name=solr6
18         -Dshare.host=localhost
19         -Dalfresco.port=8082
20         -Daos.baseUrlOverwrite=http://localhost:8082/alfresco/
21         aos
22         -Dmessaging.broker.url=\"failover:(nio://activemq:61616)
23         ?timeout=3000&jms.useCompression=true\"
24         -Ddeployment.method=DOCKER_COMPOSE
25         -Dcsrf.filter.enabled=false
26         -Xms1g -Xmx1g
27         -Dauthentication.chain=identity-service1:identity-
28         service,alfrescoNtlm1:alfrescoNtlm
29         # KeyCloak
30         -Didentity-service.enable-basic-auth=true
31         -Didentity-service.authentication.validation.failure.
```

```

    silent=false
29     -Didentity-service.auth-server-url=https://example.com/
        auth
30     -Didentity-service.realm=alfresco
31     -Didentity-service.resource=activiti
32     -Didentity-service.public-client=true
33     -Didentity-service.ssl-required=none
34     "
35     ports:
36         - 8080 #Browser port
37     volumes:
38         - ./data/alf-repo-data:/usr/local/tomcat/alf_data/
39     restart: always
40
41     share:
42         #image: alfresco/alfresco-share:6.1.0
43         build: ./share/
44         #mem_limit: 1g
45         environment:
46             - REPO_HOST=alfresco
47             - REPO_PORT=8080
48             - "CATALINA_OPTS= -Xms500m -Xmx500m"
49         ports:
50             - 8080
51         restart: always
52         links:
53             - "alfresco:alfresco"
54
55     postgres:
56         image: postgres:10.1
57         #mem_limit: 1500m
58         environment:
59             - POSTGRES_PASSWORD=alfresco
```

```
60         - POSTGRES_USER=alfresco
61         - POSTGRES_DB=alfresco
62         - PGDATA=/var/lib/postgresql/data/pgdata
63     command: postgres -c max_connections=300 -c log_min_messages=LOG
64     ports:
65         - 5432
66     volumes:
67         - ./data/postgres-data:/var/lib/postgresql/data/
68     restart: always
69
70     solr6:
71         image: alfresco/alfresco-search-services:1.3.0-RC2
72         #mem_limit: 2500m
73         environment:
74             #Solr needs to know how to register itself with Alfresco
75             - SOLR_ALFRESCO_HOST=alfresco
76             - SOLR_ALFRESCO_PORT=8080
77             #Alfresco needs to know how to call solr
78             - SOLR_SOLR_HOST=solr6
79             - SOLR_SOLR_PORT=8983
80             #Create the default alfresco and archive cores
81             - SOLR_CREATE_ALFRESCO_DEFAULTS=alfresco , archive
82             - "SOLR_JAVA_MEM=-Xms2g -Xmx2g"
83         ports:
84             - 8983 #Browser port
85         volumes:
86             - ./data/solr-data:/opt/alfresco-search-services/data/
87         restart: always
88
89     activemq:
90         image: alfresco/alfresco-activemq:5.15.8
91         #mem_limit: 2048m
92         ports:
```

```
93         - 8161 # Web Console
94         - 5672 # AMQP
95         - 61616 # OpenWire
96         - 61613 # STOMP
97     restart: always
98
99     proxy:
100         image: nginx
101         volumes:
102             - ./nginx:/etc/nginx/conf.d/
103         ports:
104             - 9999:80
105         restart: always
106         environment:
107             - NGINX_HOST=example.com
108             - NGINX_PORT=80
109
110     process:
111         build: ./process/
112         environment:
113             ACTIVITI_DATASOURCE_USERNAME: alfresco
114             ACTIVITI_DATASOURCE_PASSWORD: alfresco
115             ACTIVITI_CSRF_DISABLED: 'true'
116             ACTIVITI_CORS_ENABLED: 'true'
117             ACTIVITI_ES_SERVER_TYPE: client
118             ACTIVITI_ES_DISCOVERY_HOSTS: elasticsearch:9300
119             ACTIVITI_ES_CLUSTER_NAME: elasticsearch
120         volumes:
121             - "~/.activiti/enterprise-license:/root/.activiti/enterprise-
122               license/:ro"
123             - ./data/activiti-app.properties:/usr/share/tomcat/lib/
124               activiti-app.properties
125             - ./data/transform.lic:/usr/share/tomcat/lib/transform.lic
```

```
124     - ./ data / psdata : / usr / local / data /
125     ports :
126     - 8900:8080
127     links :
128     - elasticsearch : elasticsearch
129     - postgres : postgres
130     depends_on :
131     - elasticsearch
132     - postgres
133
134     process-admin :
135     image : alfresco / process - services - admin
136     ports :
137     - 8080
138
139     elasticsearch :
140     image : elasticsearch : 1.7.3
```

#### Quelltext 15: Vollständiges docker-compose File mit gesamter Alfresco-Umgebung

Die angegebenen Speicherlimitierungen sind Empfehlungen vom Hersteller. Im Rahmen dieser Arbeit konnten sie weder verifiziert, noch widerlegt werden, da keine Auslastung unter Realbedingungen erreicht werden konnte. Die Werte sind während des Betriebs zu prüfen und ggf. anzupassen. Weiterhin sind diese Optionen nur in der docker-compose Version 2 nutzbar. Ab Version 3 entfallen diese Angaben.

Beispielhaft wird ein komplettes Setup gezeigt. Die Container `solr6`, `activemq`, `proxy`, `process`, `process-admin` und `elasticsearch` sind für einen Betrieb nicht zwingend erforderlich. Entsprechend würden jedoch Funktionen fehlen, die die Container bereitstellen.

---

## Literatur

- [GSSZ99] Jürgen Gulbins, Markus Seyfried und Hans Strack-Zimmermann. *Dokumenten-Management: Vom Imaging zum Business-Dokument ; mit 26 Tabellen*. 2., überarb. und erw. Aufl. Berlin: Springer, 1999. ISBN: 3540615954.
- [Jeg02] Marcus Jegszent. "Verringerter Bürokratieaufwand durch effektives Dokumentenmanagement". Masterarbeit. Merseburg: HS Merseburg, 9.10.2013. Online verfügbar unter: <http://edoc2.bibliothek.uni-halle.de/oa/urn/urn:nbn:de:gbv:542-479>.
- [Alf06] Inc Alfresco Software. *Release Candidate 1 (Oct 2005)*. 6.06.2015. Online verfügbar unter: <https://community.alfresco.com/docs/DOC-4779-release-candidate-1-oct-2005>.
- [Alfg] Inc Alfresco Software. *Simply a better way to create amazing digital experiences*. Online verfügbar unter: <https://www.alfresco.com/>.
- [rwe05] rwetherall. *what other scripting languages will Alfresco support?* 2005. Online verfügbar unter: <https://community.alfresco.com/thread/142483-what-other-scripting-languages-will-alfresco-support#comment-591857>.
- [For19] Inc. Formtek. *The Alfresco Digital Business Platform*. 2019. Online verfügbar unter: <https://formtek.com/products/alfresco/>.
- [Alfe] Inc Alfresco Software. *Overview*. Online verfügbar unter: <http://docs.alfresco.com/4.0/concepts/system-about.html>.
- [Ste06] Stefan Luber / Nico Litzel. *Was ist SolR? Definition*. 27.06.18. Online verfügbar unter: <https://www.bigdata-insider.de/was-ist-solr-a-728279/>.
- [Fre17] Free Software Foundation, Inc., Joerg Kohne. *Warum man die Lesser GPL nicht für die nächste Bibliothek verwenden sollte*. 2017/04/07 12:00:20. Online verfügbar unter: <http://www.gnu.org/licenses/why-not-lgpl>.

- 
- [res06] resplin. *Open Source Licensing*. 6.06.2015. Online verfügbar unter: <https://community.alfresco.com/docs/DOC-4689-open-source-licensing>.
- [Alfa] Inc Alfresco Software, Hrsg. *Alfresco Enterprise Edition vs. Community Edition*.
- [Alf19a] Inc Alfresco Software. *Alfresco Community Editions*. 2019. Online verfügbar unter: <https://www.alfresco.com/de/ecm-software/alfresco-community-editions>.
- [alf19] alfres. *Preisgestaltung speziell für Sie?* 2019. Online verfügbar unter: <https://www.alfresco.com/de/plattform/preise>.
- [Hol15] Holger Reibold. *Alfresco 5.0 kompakt: Einstieg in das Dokumentenmanagement mit der freien Community Edition*. Enterprise.Edition. Saarbrücken: Brain-Media, 2015. ISBN: 978-3-95444-210-2.
- [Aim07] Aimprosoft. *Why do you need a document management system and how to create one?* 19.07.2018. Online verfügbar unter: <https://www.aimprosoft.com/blog/need-document-management-system/>.
- [Jac07] Jack Wallen. *How to build a Docker Compose file*. 23.07.2019. Online verfügbar unter: <https://www.techrepublic.com/article/how-to-build-a-docker-compose-file/>.
- [Red] RedHat. *Was ist Docker?* Online verfügbar unter: <https://www.redhat.com/de/topics/containers/what-is-docker> (Zuletzt geprüft am 7. Mai 2019).
- [Ges10] Geshan Manandhar. *Why use docker? 3 reasons from a development perspective*. 12.10.2018. Online verfügbar unter: <https://dev.to/geshan/why-use-docker-3-reasons-from-a-development-perspective-2jh3> (Zuletzt geprüft am 7. Mai 2019).
- [Alf19c] Inc Alfresco Software. *Installing alfresco-pdf-renderer*. 2019. Online verfügbar unter: <https://docs.alfresco.com/5.2/tasks/pdf-renderer-install.html>.

- 
- [Alf19d] Inc Alfresco Software. *Installing ImageMagick*. 2019. Online verfügbar unter: <http://docs.alfresco.com/5.2/tasks/imagemagick-config.html>.
- [Alf19e] Inc Alfresco Software. *Installing LibreOffice*. 2019. Online verfügbar unter: <http://docs.alfresco.com/5.2/tasks/lo-install.html>.
- [Alf19b] Inc Alfresco Software. *Deploying using Docker Compose*. 2019. Online verfügbar unter: <https://docs.alfresco.com/6.0/tasks/deploy-docker-compose.html>.
- [Alfd] Inc Alfresco Software. *Configuring Share with the share-configcustom.xml file*. Online verfügbar unter: <https://docs.alfresco.com/5.0/tasks/share-customizing-custom-config-file.html>.
- [Sof] SoftSelect. *Definition Add-On*. Online verfügbar unter: <http://www.softselect.de/business-software-glossar/add-on>.
- [pre19] pressengers. *Schadcode in WordPress Plugin „Display Widgets“ gefunden: Backdoor für Fernzugriff gefunden*. 2019. Online verfügbar unter: <https://pressengers.de/plugins/schadcode-in-wordpress-plugin-display-widgets-gefunden/>.
- [ino18] inorton. *Why we took AddOns down - and whats happening next*. 2018. Online verfügbar unter: <https://community.alfresco.com/community/ecm/blog/2018/05/29/why-we-took-addons-down>.
- [Alff] Inc Alfresco Software. *Protocols*. Online verfügbar unter: <https://docs.alfresco.com/4.2/concepts/protocols-about.html>.
- [Key] KeyCloak. *About*. Online verfügbar unter: <https://www.keycloak.org/about.html>.
- [Die11] Die Druckerei. *Corporate Design: Kompakt-Guide von der Konzeption zum Corporate Design Manual*. 29.11.2018. Online verfügbar unter: <https://www.diedruckerei.de/magazin/corporate-design-manual-konzeption/>.

- [Alfc] Inc Alfresco Software. *Benutzerrollen und Berechtigungen*. Online verfügbar unter: [https://docs.alfresco.com/using-de/references/permissions\\_share.html](https://docs.alfresco.com/using-de/references/permissions_share.html).
- [For] Forschungsstelle Recht im DFN. *Handlungsempfehlung zur Öffnung von E-Mail-Accounts und Benutzerkonten*. Online verfügbar unter: [https://www.dfn.de/fileadmin/3Beratung/Recht/handlungsempfehlungen/Oeffnung\\_von\\_E-Mail-Accounts\\_und\\_Benutzerkonten.pdf](https://www.dfn.de/fileadmin/3Beratung/Recht/handlungsempfehlungen/Oeffnung_von_E-Mail-Accounts_und_Benutzerkonten.pdf).
- [Alfb] Inc Alfresco Software. *Alfresco super users*. Online verfügbar unter: <https://docs.alfresco.com/community5.0/topics/alfresco-superusers.html>.
- [Nor17] Norbert Kloiber. *Backup erstellen – was muss man beachten? Backup erstellen – ein unbedingtes Muss!* 2017. Online verfügbar unter: <https://start.norbert-kloiber.at/backup-erstellen-was-muss-man-beachten/>.
- [Piv19] Pivotal Software. *Spring Framework*. 2019. Online verfügbar unter: <https://spring.io/projects/spring-framework>.
- [Bos] Bosch Software Innovations. *VFS Connector*. Online verfügbar unter: [http://documentation.bosch-si.com/bpm/7\\_0/online-help/de/Plugin.VFSConnector.html](http://documentation.bosch-si.com/bpm/7_0/online-help/de/Plugin.VFSConnector.html).

## Selbstständigkeitserklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Merseburg, den 28. August 2019

-----  
Kevin Wagner