Funktion und Architektur

SQUEEZE ist eine Mandanten-fähige Software-Lösung für Dokumenten-Klassifikation und Inhaltsextraktion. Dieses Dokument beschreibt die technischen Leistungsmerkmale von SQUEEZE und beschreibt vorrangig die technische Architektur, Eigenschaften und Funktionen.

SQUEEZE wurde im Jahr 2016 designed und aufgebaut und wird seither stetig in einem wachsenden Team weiterentwickelt.

Es ist ein vollständig webbasiertes System zur Verarbeitung von elektronischen Dokumenten, mit wesentlichem Fokus auf die Erkennung und Auslesung von Eingangsdokumenten im B2B Sektor.

Konzepte des Systemdesigns

Für das Systemdesign wurden folgende Konzepte verfolgt:

- Betriebssystem offen in Bezug auf Linux und Windows Server
- Datenbank-offen in Bezug auf mySQL/MariaDB und Microsoft SQL Server
- Konsequent webbasierter Administrations- und Anwendungszugang
- Vollständige REST-API auf Swagger.io / OpenAPI v3
- Schnelle und problemfreie Installation bei OnPrem-Projekten sowie Update-Fähigkeit
- Wartungsarmer Betrieb, interne Überwachung / Monitoring
- Hohe I in Dritt-Anwendungen
- Einfache Konfiguration über das Webinterface für Partner und Administratoren
- Hohe Stabilität in Bezug auf die Verfügbarkeit
- Hohe Verarbeitungsgeschwindigkeit und Erkennungsqualität

Systemarchitektur

SQUEEZE Server

SQUEEZE ist vorrangig eine HTTP-Applikation auf PHP-Basis, welche im Apache Webserver gehostet wird und die serverseitige Verarbeitung steuert. Weiterhin stellt SQUEEZE einen Webclient für die Administration und Endanwendernutzung zur Verfügung.

SQUEEZE Worker

Worker verstehen sich als eigenständige, asynchron agierende Prozesse, welche Subprozesse (z.B. OCR-Vorgänge) des Systems veranlassen. Das Worker-Konzept bietet eine flexible Skalierung aller Systemarbeitsprozesse (Bildaufbereitung, OCR, Klassifikation, Extraktion) und sind ein wesentliches Element der SQUEEZE Arbeitsweise.

Innerhalb der Software können so viele Worker aktiviert werden, wie lizenziert sind. Die kleinste Standard-Lizenz umfasst 4 Worker. Die maximale Anzahl gleichzeitiger Worker-Prozesse orientiert sich an der Anzahl der Prozessorkerne, über die das Betriebssystem verfügt, respektive sollte der Anzahl der CPU Threads nicht übersteigen. Erfolgreich getestet wurde das System mit bis zu 64 Threads.

Die Worker auch auf anderen Servern zu verteilen, wird in der aktuellen Auslieferung nicht empfohlen auch wenn dies theoretisch möglich wäre. Die Schreibvorgänge der Datenbank können, je nach Ausstattung, an die Grenzen kommen. Hierzu wurden noch nicht ausreichende Tests durchgeführt.

SQUEEZE Datenbank

Die Squeeze Datenbank enthält die Konfigurationen des Systems sowie die Nutzdaten der Kunden.

Bei der Datenbank handelt es sich um eine relationale Datenbank (siehe Systemvoraussetzungen)

SQUEEZE Message Queue

Die SQUEEZE Message Queue nutzt RabbitMQ um die Aufgaben der Worker zu verwalten. Die Worker registrieren sich als Konsument bei der Message Queue und arbeiten die Messages ab, sobald sie Zeit dafür haben.

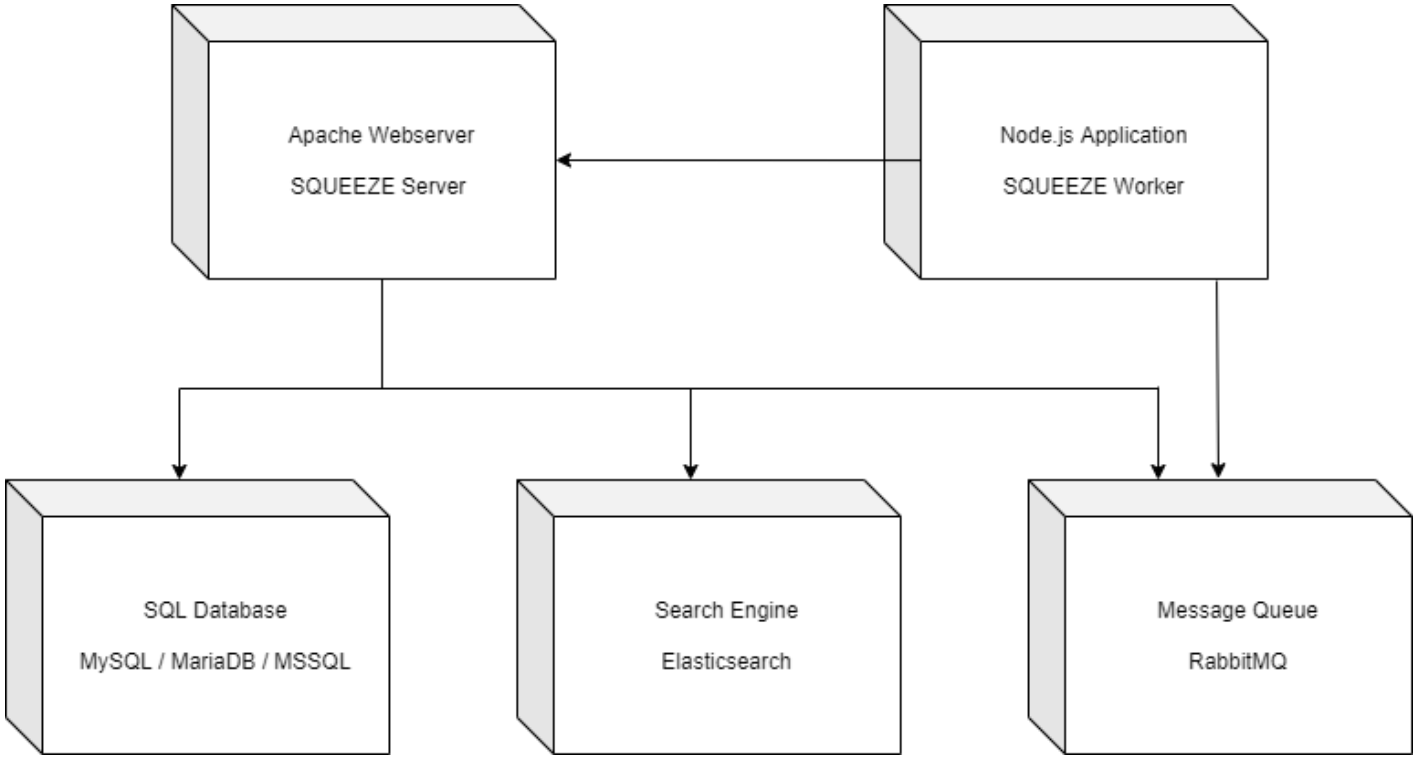
SQUEEZE Repository

Das SQUEEZE Repository enthält alle eingelesenen Belege / Dokumente. Aktuell werden die Dokumente als Datei auf der Festplatte des SQUEEZE Servers gespeichert. Zukünftig wird es optional die Möglichkeit geben die Dokumente in einem S3 Bucket zu speichern, um auch auf dieser Ebene besser skalieren zu können.

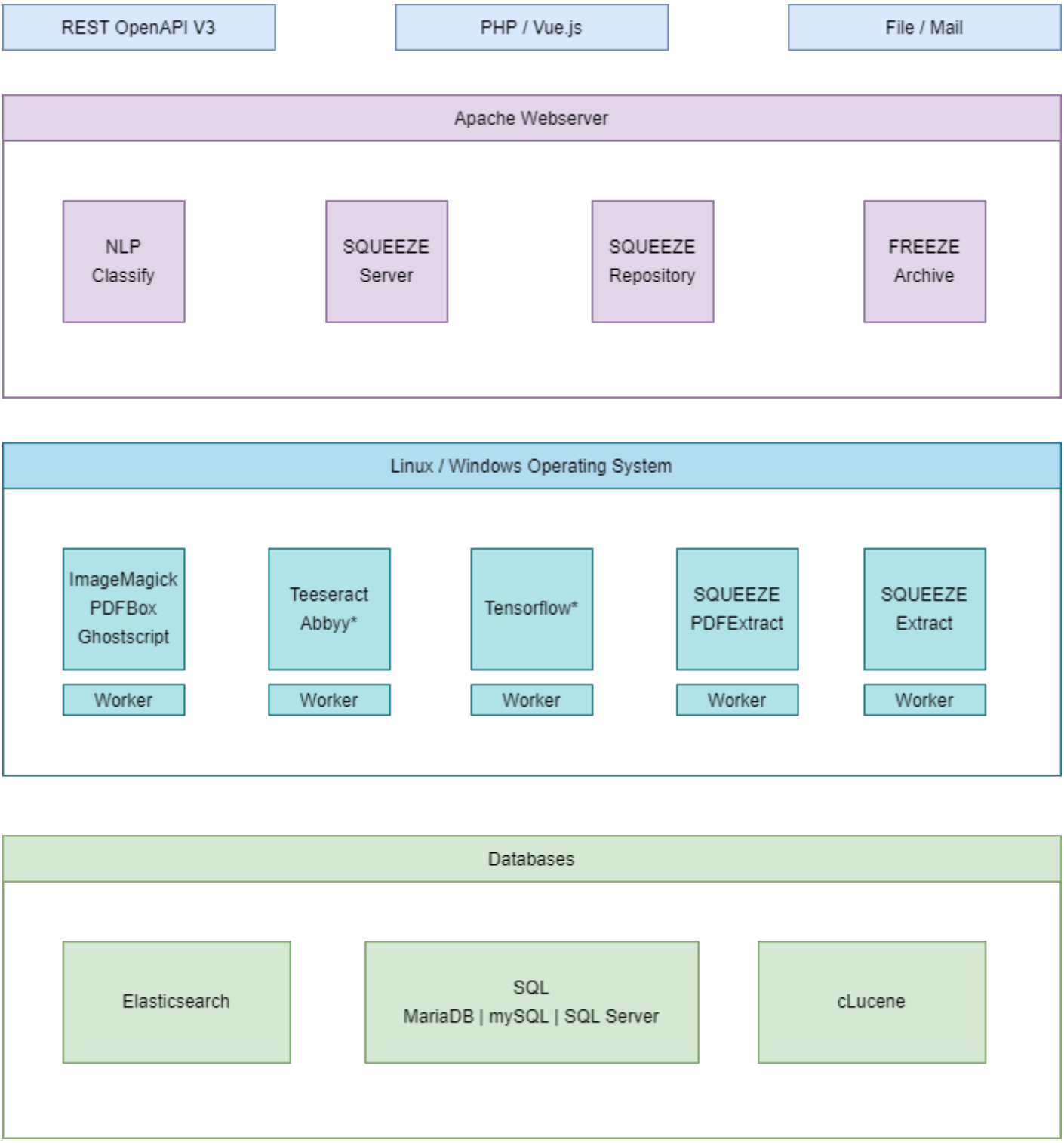
SQUEEZE Volltext

SQUEEZE arbeitet mit Elasticsearch. Dies ist eine Volltextsuchengine und stellt phonetische, linguistische und unscharfe Suchmethoden zur Verfügung. Zukünftig dient es zusätzlich als Datenlieferant für Unternehmensstammdaten in erweiterten Konfigurationen (Mailroom Szenarien).

SQUEEZE Systemabbildung



SQUEEZE Stack



Revision #14
Created 2020-03-03 14:44:58 UTC by Phillip Langer
Updated 2020-03-25 15:22:46 UTC by Phillip Langer