
AIP_03: “Architekturentwurf”

Felix Baumgartner & Jonas Seegers

Agenda

- Userbeschreibungen
 - Umgebende Systeme
 - Geforderte Funktionen
 - Bausteinsicht
 - Kontextsicht
 - Verteilungssicht
 - Verwendbare Frameworks
 - Fragen
-

Userbeschreibungen

Servicekraft

- Greift einmal wöchentlich auf seinen Plan zu (Webinterface oder Smartphone App über REST-API)
 - Wird per Mail über Änderungen informiert
 - Hat ein User und Passwort
 - 150 Servicekräfte + 10-20/Jahr
-

Planer

- Erstellt einmal pro Woche neue Pläne
 - Nur zu den normalen Bürozeiten
 - 1-3 Planer
-

Umgebende Systeme

SugarCRM

- Kundendaten
 - Anlagendaten
 - Wartungstermine (inklusive Anfahrt und Dauer)
 - oauth2
 - Daten werden per REST API abgefragt
-

MySQL Mitarbeiterdatenbank

- Verfügbarkeit der Mitarbeiter
- Puffer für Notfälle

ORM

- Einfache Handhabung mit Objekten
- Alle Entitäten die wir benutzen lassen sich gut in Objekte parsen
- Verschachtelung ist möglich und gefordert

Falls ORM nicht gewünscht Wrapper möglich

Geforderte Funktionen

Webinterface für Planabholung

- nur lesender Zugriff auf Pläne
 - Abfrage von User / Passwort
 - REST API
 - Verfügbarkeit mindestens 99,9%
 - Verteiltes System
 - Ringarchitektur
-

REST API

- Zugriff für Apps
 - Zugriff für das Webinterface
 - oauth2
-

Planerstellungstool

- Webanwendung
 - Teilautomatisierte Erstellung der Pläne
 - Verfügbarkeit der Servicekräfte (MySQL)
 - Anstehende Wartungstermine (CRM)
 - Puffer für Notfälle (MySQL)
 - Nach Freigabe durch den Planer E-Mail an betreffende Servicekraft
 - REST API
-

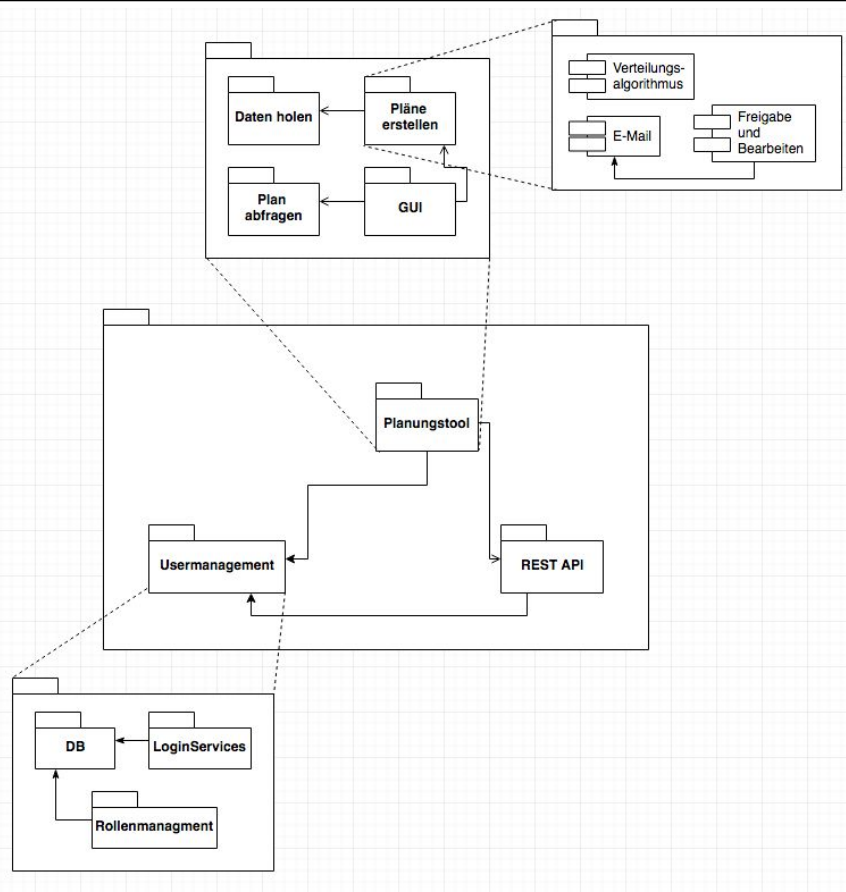
Teilautomatisierte Erstellung

- Verschiede Algorithmen möglich:
 - Blackboard- Agenten (Performanter, einfacher, mehr Overhead für Planer)
 - Jeder Servicekraft wird ein Agent zugeteilt
 - Blackboard mit Terminen
 - Queue mit Agenten
 - Termin nehmen wieder anstellen
 - Verteilungsserver (Langsamer, komplizierter, weniger Overhead)
 - Liste aller Termine
 - Liste aller Kalender
 - Für jeden Termin den am Besten passenden Kalender wählen
-

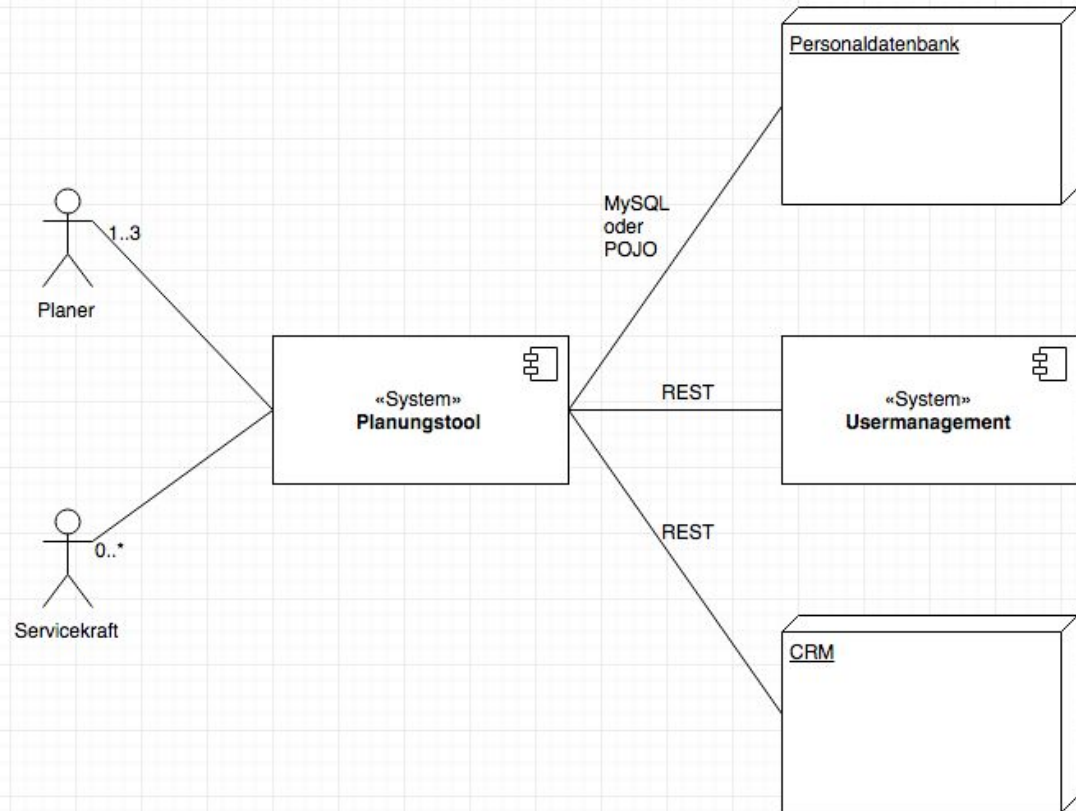
Usermanagement

- Microservice mit Framework
 - Framework wird erweitert
 - Eigene kleine DB
 - Username
 - Passwort (Verschlüsselt)
 - Rolle
 - Personalnummer
 - Benutzerrollen mit entsprechenden Rechten
-

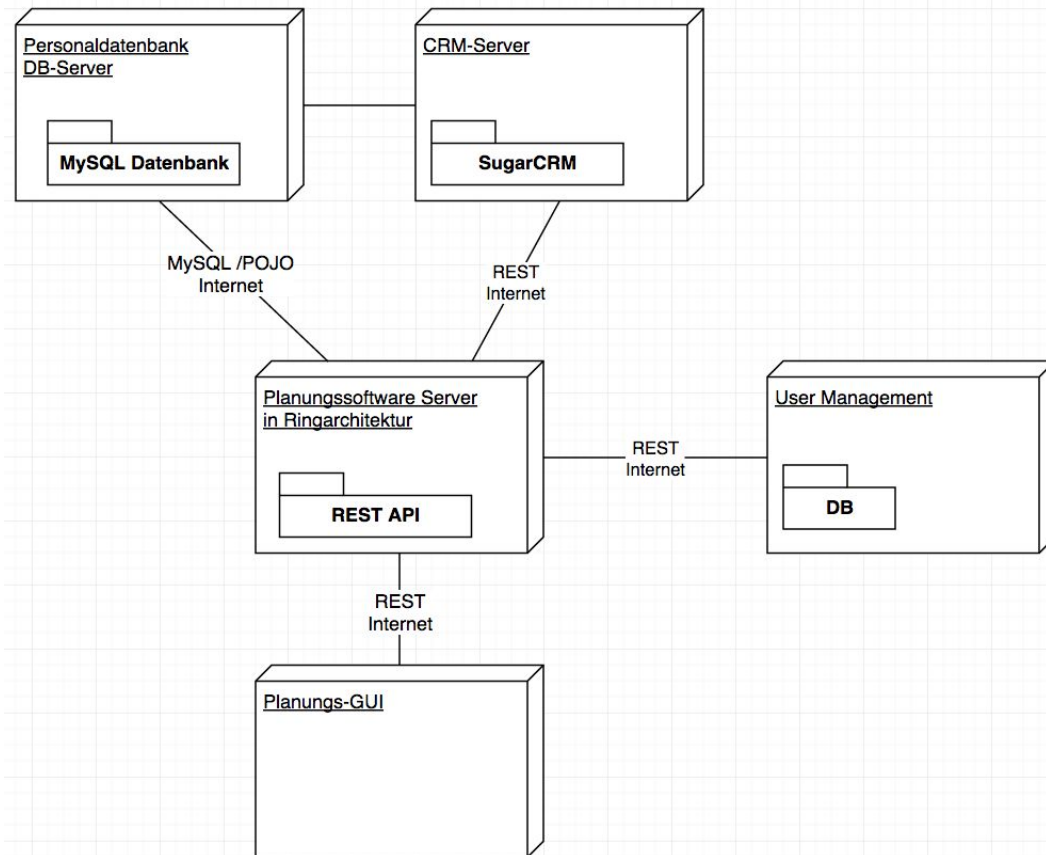
Bausteinsicht



Kontextsicht



Verteilungssicht



Verwendbare Frameworks

Frameworks

- Stormpath (Usermanagement)(node.js)
 - UserFrosting (Usermanagement)(php)
 - Hibernate (ORM)
-

Fragen

Fragen

- Wie viel händischer Aufwand soll vom Planer kommen?
 - Wie lange ist ein Durchschnittstermin?
 - Wie viele lange Termine und wie viele kurze Termine hat man pro Woche?
-

Ende
