Betreff: PAC 2019 Task 1.0

Datum: Sonntag, 23. Februar 2020 um 15:43:02 Mitteleuropäische Normalzeit

Von: Darko Križić

An: Andreas Heizenreder, Armin Aria, Fabian Siems, Bünyamin Sever, Jovan Milosevic, Matti Mäkitalo, Paul Schell, Ralf Janiak, Yusuf Can Kesen

Kategorie: Prodyna

PAC 2019 Task 1.0

Abstract

The Conferencing App is a fictive but easy to understand business application. The customers organises events and wants to get rid of printed paper and digitise the business. The intention is to have a central place where all details about events, talks, persons a.s.o. are stored. The system must support multiple frontends (e.g. Web or Mobile phone). The system must be scalable because while the events take place the load gets very high. So the architecture must compensate for that. Additionally monitoring must ensure that bottlenecks can be detected early to prevent low performance.

Requirements

- The solution must work in CNEE
  - This implies Docker and Kubernetes
  - You can use provided Jenkins of CNEE
- The solution must offer a dedicated headless backend
  - The backend offers full CRUD functionalities for all entities
  - The backend allows anonymous read operations
  - All other operations are only allowed for authenticated users
  - It exposes all data via a standard interface (e.g. REST or GraphQL)
- The solution must offer a frontend
  - It accesses the backend via the standard interface in a secure way (e.g. REST via HTTPS)
  - It can be any technology (e.g. Web App, Mobile App)
- The solution must be scalable to offer dynamic cost (run single and scale out)
- The solution must offer a monitoring of various KPIs
  - Request times
  - Request count
  - Memory consumption
  - CPU consumption

Mainstream technologies

The following technologies are considered main stream:

- Java, Java EE, Spring Boot
- JavaScript, Node, Angular, React
- RDBMS, Neo4j
- REST
- Prometheus, Grafana

Shortcuts

The frontend does not have to implement all operations of the backend. What needs to be done is:

- Implementation of all the views described below

- Authentication of the user
- One (or more) write operations (e.g. adding a topic, renaming a talk)

Entity Model

- A **Location** describes a location where Events take place
    - It has a name
- An **Event** describes an event
    - It is in a **Location**
    - It has a begin and end date
- An **Organization** describes a company where a person belongs to
    - It has a name
- A **Person** describes a person
    - It has a name
    - It belongs to an **Organization**
- A **Room** describes a room where talks are held
    - It has a name
    - It belongs to an **Organization**
- A **Talk** describes a talk
    - It has a title
    - It has a duration
    - It has a **Language**
    - It is held by one or more **Persons**
    - It has a level (e.g. beginner, advanced, expert)
    - It has one or more Topics
    - Takes place in a combination of the following
        - A **Room** in the **Event** of the **Location**
        - A specific time. Note that a room can be only allocated for one Talk for the talk duration.
- A **Topic** describes a topic that are used in talks
    - It has a name
    - It can have any number of children
    - It can have any number of parents
    - In summary: Topics are a directed graph

Views

The following views need to be implemented

- A **list of all Events**
    - Start and end date
    - **Location**
    - **Topics** of the talks in the event
- A d**ay overview** for a specific event
    - Tabular view of the whole day
        - Horizontal: The rooms
        - Vertical: The time, ordered by time
    - Each entry shows
        - The title
        - The topic(s)
        - The person(s)
        - The level
- A list of Persons
- A **Person** overview
    - Contains a list of all **Talks** of the Person
- A **Talk** overview
    - Contains the title

  - The **Level**
  - The **Language**
  - The **Persons**
  - The **Topics**
  - The **Events** where it is held

Target Environment

The Application must run in the CNEE, this implies:

- It is Kubernetes
- CNEE provides a namespace for the application
- The artefacts must Docker images

Find information about CNEE here:
https://secure.prodyna.de/confluence/display/PDTED/CNEE+Quickstart

Milestones

- Second block
  - You present the approach of your solution
  - Suggestion: Create a PowerPoint presentation that addresses all important topics
  - Time: 30 minutes + 15 minutes discussion and feedback
- Two weeks before last block
  - You provide a Git Repository with all source code (for review)
- Last block
  - You present the final solution
  - The running Application in CNEE
  - Suggestion: Take your existing approach presentation and extend it with details about your solution
  - Time: 60 minutes + 30 minutes discussion and feedback
  -