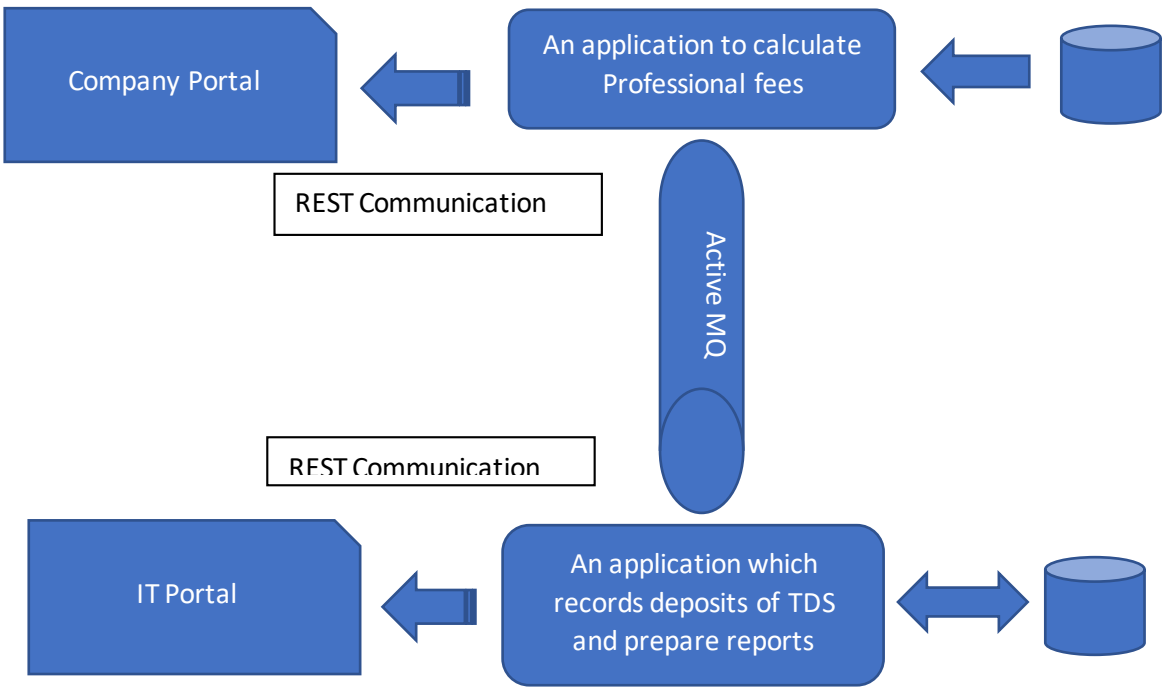
Problem Statement

An organization has around 1000 professionals on board. They are looking for the professional fees calculation system which will also calculate Tax Deduction at Source. The TDS will be deposited to another TDS application which on receipt of TDS will record it through which will allow finance department to view through reports.



There are two applications to be designed as micro services.

Application 1: An application to calculate monthly professional fees from the yearly fees of each employee recorded in Data Base. This calculation has to happen when manually triggered by Authority of Finance Department. The application also to calculate TDS and send the TDS details of each employee to the Application 2 through Active MQ.

Application 2: An application to record every TDS deposited to the Data Base for every employee. The TDS deposited must be identified using PAN Number of employee and Month-Year for which it being deposited.

Technologies and Frameworks to use:

1. Spring Cloud, Spring Boot, Spring REST, Angular 6, Data JPA, JMS-Active MQ, Discovery using Eureka, Load Balancing using Ribbon, Failover and circuit breaking using Hystrix,
2. Tools: Oracle Database, STS or Eclipse with Spring Boot plugin.

Assumptions:

The security, authentication and authorization are out of scope of the problem statement.

Phase-I

Application 1: Actors- Finance Department

1. Get the Employee details including salary details of a given employee.
2. Calculate monthly salary for every employee and deposit TDS (10% of Salary) to the IT application.

Table Structure: EmpId, EmpDetails-Name-Address, Email, PAN, Annual Package.

Use Case 1 is the REST services published and are accessible by Angular client.

Use Case 2 is a publisher of TDS data for JMS.

Application 2: Actors- Finance Department and Application 1

3. Insert new TDS deposit record to the data base for each employee each month.
4. Get the list of TDS paid for a given tax payer.

Table Structure: PAN, Month, Year, TDS Amount, email, Employee Name.

Use Case 1: It's a subscriber of TDS data from JMS.

Use Cases 2 is the REST services published and are accessible by Angular Client.

Design the microservices, discover them using Eureka. Load Balance them using Ribbon and design fail-over mechanism using Hystrix.

Phase II:

Application 1: Actors- Finance Department

5. Get the list of the Employee Ids.

Use Case 5 is the REST services published and to be accessible by Angular client.

Modify the Angular client to populate the list of EmpIds into a combo box. User when selects an employee id, the url should refer to Use Case 1.

Application 2: Actors- Finance Department and Application 1

6. Get the list of PAN Numbers

Use Case 6: It's a subscriber of TDS data from JMS.

Modify the Angular client to populate the list of PAN numbers into a combo box. User when selects PAN number, the url should refer to Use Case 3.

Modify the application to prepare pay slip in PDF.

Modify the application to notify tax payers through mail.

Scheduler to calculate salary on 1st of every month.