BACK-END ASSIGNMENT #1

General Information

A Private Branch Exchange (PBX) is a telephony system in an organization that enables internal and external communication through the phone. Usually, any internal communication is done over the organization's private network and external through the internet using Voice over IP (VoIP). Such a system can also be hosted online (in a private or public cloud) and is referred to as cloud PBX. Some of its basic features include call transfers, voicemail, call recording, interactive voice menus (IVRs), and call queues. Additionally, a cloud PBX provides an API and web-hooks. Any communication (call) in a PBX has a record that is usually referred to as Call Detailed Record (CDR).

Problem / Challenge

In a micro-service environment, the need to introduce a new micro-service was raised after customers started asking to have their phone call records available in their CRM. For this reason, the engineering team has to create a standalone micro-service that can integrate with multiple Telecom systems (PBX) and store the call records after matching them with the phone-book directory.

The integration will utilize both the CDRs API and web-hook calls to retrieve CDRs from any cloud PBX that might be used by a customer. The CDRs API is required is to fetch the historical data and to keep the records synchronized while the web-hooks will store the CDRs while a new call is made. For simplicity reasons, you are provided with two files that have mock data & instructions for both CDR API & web-hooks.

As mentioned in the first paragraph, in addition to the integration, the engineering team must also implement a phone-book directory to the same micro-service. Before storing any CDR record, this micro-service will use the directory to match the record with the relevant phone-book entry based on the number.

For this assignment, a mock PBX API application was developed. The provided compressed file contains the Node.js application and a .csv file (pb_directory.csv) with sample contacts. Extract the file follow the instructions under the README.MD file.

Requirements

- Use Spring Boot 2
- Implement the endpoints & methods described in Table 1
- Implement any additional endpoints & methods
- Design & Describe the database (Use PostgreSQL)
- Use best-practice patterns & approaches for micro-services production-ready architecture

Table 1

DESCRIPTION	METHOD	URI
CRUD operations for PBX integration	POST, GET, PUT, DELETE	Not provided
CRUD operations for CDR records	POST, GET, PUT, DELETE	Not provided
CRUD operations for phone-book directory	POST, GET, PUT, DELETE	Not provided

A method that consumes CDR API and updates the records accordingly. Used when a new integration is added and for sync purposes at the end of each day.	N/A	Not provided
End-point that consumes all the calls made by web-hooks (PBX) and updates CDR records accordingly.	POST	Not provided
An endpoint that provides a detailed report on the CDR data of each tenant. Include the following: Total number of calls (CDRs) Total number of minutes of successful calls incomplete calls Total number of unmatched calls Latest sync date	GET	Not provided

Notes:

- Each incoming request passed through an API Gateway and was authorized to proceed
- Each request contains the customer UUID in its headers
- The CDRs API & web-hook JSON formats are always the same (for simplicity)

Submission

Please submit your solution through email. The submission must include:

- Source Code archive
- Database design & implementation
- Docker compose file
- Document with any assumptions made & project guidelines in a README.MD file

Validation

Your submission will be evaluated based on the following criteria:

- Working solution that implements all endpoints described in the requirements
- Code quality, comments and readable internal structure
- Tests & code coverage