Synapsis

Back End Engineer Challenge

PT Synapsis Sinergi Digital

The purpose of this assignment is to test your skills in software engineering, specifically back-end development. Task expectations are described below.

A. Objective:

Design and implement a backend system with microservice architecture using RPC (e.g., HTTP or gRPC) for interservice communication. You should focus on clear service boundaries, scalability, and code quality. You don't need to create a front-end application, just focus on creating a RESTful API.

B. Technical Specification:

- Programming Language : Go
- Database : PostgreSQL
- System Architecture : Microservices with Choreography Pattern
- Interservice Communication : gRPC
- Mocking Library (for unit test) : mockery v2 or higher

C. Requirements:

Build **at least 2 core services**:

1) Order-service
    a) Accepts orders via REST.
    b) Validates inventory availability (via RPC call to inventory-service).
    c) Responds with order status (confirmed or rejected).
2) Inventory-service
    a) Manages product stock.

b) Exposes RPC endpoints to:
   i) Check stock
   ii) Reserve stock
   iii) Release stock (in case of failed order)

Optional bonus services:

3) Notification-service
   a) Sends email or logs notification when order is placed or fails.
4) User-service
   a) Authenticates users placing orders.

D. Technical Requirements:

1) Use **RPC communication** between services (gRPC).
2) Each service must be **independently runnable** (each service has its own database).
3) Implement proper **error handling** and **time-out/retry logic** for RPC calls.
4) Implement proper **locking** and **transaction** mechanisms.
5) Capture application **logs** and **errors**, centralized logs monitor will be a plus.
6) Write a unit test for **all business logic functions** (handler/service layer).
7) Document your architecture and assumptions clearly.

E. Deliverables:

1) **Codebase**: Pushed to GitHub in a single repo, organized by service (e.g., `/order-service`, `/inventory-service`)
2) **README**:
   a) Setup instructions
   b) System overview and design choices
   c) How to run and test
3) **System Design Diagram** (PNG, PDF, or in README)

4) **ERD** (PNG, PDF, or in README)
5) **Unit tests** for at least one service

F. Submission:

You must submit the challenge within **4 days** of receiving it.

Please follow these steps to complete your submission:

1) Code Repository

a) Save your completed code to a private GitHub repository.
b) Invite the user **backend-syn** (GitHub profile: https://github.com/backend-syn) to access the repository.

2) Submission Form

Submit the following links via the form we have provided:

a) The **GitHub repository link**
b) The **API specification link**

Completing the challenge ahead of the deadline will be taken into consideration during the evaluation process.

G. What We Evaluate:

1) Programming and API design
2) Clean code and service isolation
3) Reasoning behind decisions (trade-offs, limitations)
4) Understanding of system and database
5) Documentation and testing discipline

– Do Your Best –