🧩 Programming Assignment: Outbound Voice Campaign Microservice

Problem Overview

You are tasked with enhancing an existing microservice that currently provides:

1. An API to **trigger an outbound voice call**, and
2. An API to **check the status** of a call (`in_progress`, `completed`, or `failed`).

Your goal is to extend this system to support **outbound call campaigns** — allowing multiple calls to be triggered and managed together under a single campaign.

Requirements

The enhanced system should enable management of **outbound call campaigns**, where multiple calls can be triggered and monitored together under a single campaign. The following capabilities are expected to be supported:

1. Campaign Management

- The system should allow clients to **create and manage outbound call campaigns**.
- Each campaign should contain a group of phone numbers that need to be called.
- Campaigns should maintain their own lifecycle and track the progress and results of all calls within them.

2. Business Hour Scheduling

- Campaigns should support defining **specific business hours** during which calls can be made.
- Calls should be automatically scheduled and executed only within these configured time windows.

- The system should be able to handle different time zones and prevent calls from being triggered outside allowed hours.

3. Concurrency Control

- The system should allow configuration of **maximum concurrent calls** that can be active at a time within a campaign.
- If no value is specified, the system should apply a **default concurrency limit**.

4. Retry Handling

- If a call attempt fails, the system should **automatically retry** the call based on the campaign's retry configuration.
- The system should ensure that failed calls are retried before new calls are initiated, maintaining fairness and efficiency.

5. Status Tracking

- The system should track and expose the **status of individual calls** (e.g., in-progress, completed, failed) and the **overall campaign status** (e.g., pending, in-progress, completed, failed).
- Campaigns should provide aggregated statistics such as total calls, calls completed, calls failed, and retries attempted.

System Design Requirements

Apart from the code that implements the above requirements, you must include a **high-level architecture diagram** or **text-based explanation** describing:

- How APIs, databases, and background workers interact to meet the above capabilities
- The choice of technology for these **external components** (e.g., message queues like RabbitMQ, databases like PostgreSQL, task schedulers like Celery, etc.)

- How the system ensures **fault tolerance** and **scalability**.

Implementation Expectations

- You may use **any programming language or framework** (Python, Go, Node.js, Java, etc.)
- Code must be **production-quality**, readable, and modular.
- Include clear instructions for how to:
    - Set up and run the service locally
    - Run sample tests or curl commands
- Include **mock implementations** for outbound call triggering (no need for real telephony integration).

Deliverables

Candidates should submit:

1. Source code (in a Git repo or zip file)
2. README file with:
    - Setup instructions
    - Example API usage
    - System design explanation
3. (Optional) Diagrams or flowcharts for the system design