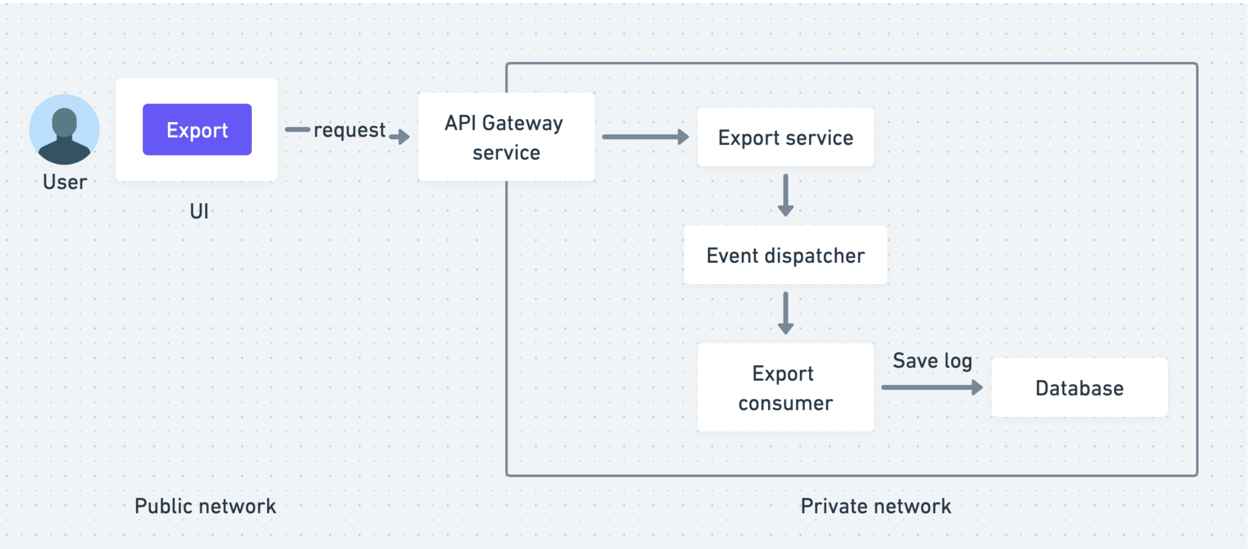
Engineering code challenge - CSV Export service

Scenario

- User can export data in a CSV file on UI, depends the data amount, it will cost 5 ~ 10 seconds to get the file.
- Only authorized user API requests should be processed. Unauthorized requests should receive an error response immediately.

Diagram



User Flow

- When **User** click export button on UI, wait 5 ~ 10 seconds, then get downloaded CSV file.

Requirement

1. Exported CSV file should include: (block 5 ~ 10 seconds)
    - Export request timestamp
    - Export completion timestamp
2. Microservices Architecture:
    - Create a microservices-based system with the following services:
        - API Gateway Service: Handles user requests and routes them to appropriate internal services.
        - Export Service: Processes export requests and generates CSV files.
        - Export Consumer: Consumes export requests and manages file delivery.

- All services, excluding the **Event dispatcher** and **Database**, should be implemented in **Go**:
- Authorization
    - Only authorized user API requests should be processed. Unauthorized requests should receive an error response immediately.
- Containerize all services using Docker, ensuring they can be run seamlessly in a local development environment.

System Goal

1. Non-blocking APIs: Utilize asynchronous processing to ensure APIs don't block each other.
2. Security: Expose only the API gateway, keeping internal services secure.
3. Extensible Export Service: Design with modularity to support various templates, datasets, or files.
4. Scalability: Implement architecture that enables effortless horizontal scaling to handle increased demand.

Nice to have

1. Auto Testing: Implement CI/CD pipeline to automatically run tests on opening a Pull Request.

2. Auto Deploy: Trigger deployment upon merging to the main branch.
3. Data Backup: Create a script to automate periodic data backups.

Questions (describe in sentences)

- If every user have 5 credits at begin (export 1 time will consume 1 credit), every 12am will gain 1 credit. Please provide your design philosophy.
- Besides the design on the diagram, What kind of technique approach you can provide to achieve the same goal? What's Pros and Cons? Please provide the diagram if you prefer.

Q2. SQL syntax

Write a SQL to:

- Got Count of each categories of menu which muscles.name should be `muscle1` OR `muscle2`.
- Same **Menu** in rows, should only count one time.
- Expected Result:
    - training 3
    - balance 1
    - stretch 1

Dataset:

```
CREATE TABLE IF NOT EXISTS menus (
   id       SERIAL PRIMARY KEY,
   name     TEXT,
   category TEXT
);

CREATE TABLE IF NOT EXISTS muscles (
   id       SERIAL PRIMARY KEY,
   name TEXT
);
```

```
CREATE TABLE IF NOT EXISTS muscle_menus (
   id  SERIAL PRIMARY KEY,
   muscle_id INT NOT NULL,
   menu_id   INT NOT NULL,
   FOREIGN KEY (muscle_id) REFERENCES muscles(id),
   FOREIGN KEY (menu_id) REFERENCES menus(id)
);

INSERT INTO menus (name, category) VALUES ('menu1', 'training');
INSERT INTO menus (name, category) VALUES ('menu2', 'training');
INSERT INTO menus (name, category) VALUES ('menu3', 'training');
INSERT INTO menus (name, category) VALUES ('menu4', 'stretch');
INSERT INTO menus (name, category) VALUES ('menu5', 'stretch');
INSERT INTO menus (name, category) VALUES ('menu6', 'balance');

INSERT INTO muscles (name) VALUES ('muscle1');
INSERT INTO muscles (name) VALUES ('muscle2');
INSERT INTO muscles (name) VALUES ('muscle3');

INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (1, 1);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (1, 2);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (1, 3);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (1, 6);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (2, 1);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (2, 4);
INSERT INTO muscle_menus (muscle_id, menu_id) VALUES (3, 1);
```

Q3. Investigate how to implement `Upload file` though API

Please describe it.

Q4. Investigate how to implement realtime `Chat feature` though API

Please describe it.

Q5. Investigate how to implement APIs for mobile app? Will it different as web app?

Please describe it.

Q6: How do you design a background process run once per hour?

Please describe it.