Challenge: Asynchronous Service Communication

Introduction

At the API layer, a key principle is to **protect underlying services from overload** while ensuring **optimal performance**, even under high-traffic conditions. We employ an **asynchronous queueing mechanism** for communication with internal services to achieve this.

Use Case: Starting a Charging Session

A driver attempts to start a charging session at a specific station by sending an API request to a REST endpoint. The request includes unique identifiers for both the station and the driver and a **callback URL** where the final decision will be sent once processing is complete.

- **Station Identifier**: A UUID
- **Driver Identifier**: A string token of 20 to 80 characters in length. Allowed characters include:
    - Uppercase letters (`A-Z`)
    - Lowercase letters (`a-z`)
    - Digits (`0-9`)
    - Hyphen (`-`), period (`.`), underscore (`_`), and tilde (`~`).
- **Callback URL**: A valid HTTP/HTTPS endpoint is provided by the client to receive the final decision.

Flow of the Request

1. The API controller receives the request, validates the input, and immediately responds with an acknowledgment.
2. The controller asynchronously forwards the **station ID**, **driver token**, and **callback URL** to an **internal authorization service** via a queueing mechanism.
3. The internal authorization service checks an **Access Control List (ACL)** to make a decision.
4. The decision is **persisted** for debugging purposes.
5. The result (decision) is sent to the provided **callback URL** in the following format:

    Callback Payload Example:

    ```
    {
        "station_id": "123e4567-e89b-12d3-a456-426614174000",
        "driver_token": "validDriverToken123",
        "status": "allowed"  // Possible values: allowed, not_allowed, unknown, invalid
    }

    ```

6. The client can determine the outcome based on the callback response.

Response to Driver’s API Request

The API immediately responds to the client with an acknowledgment:

```
{
   "status": "accepted",
   "message": "Request is being processed asynchronously. The result will be sent to the provided callback URL."
```

```
}

```

Constraints

- The **internal authorization service** exposes only an HTTP/REST interface and must **not be invoked synchronously** by the API controller.
- If the authorization service does not respond within a predefined timeout, the driver token will default to **unknown**.
- The **callback URL** must be a valid HTTP/HTTPS endpoint in the request payload.

Boundaries

- The API controller is not responsible for **rate-limiting** or **authorizing** incoming HTTP requests.
- The station identifiers (UUIDv4) are assumed to be **valid and known**.

Notes

- You may implement the solution in **PHP**, **Python**, or **Kotlin**.
- Share your solution as a Git repository link or an archive file.
- The repository must include:
  - Instructions to run the application and test suite locally.
  - Documentation for the public API endpoint.
- Supplementary materials, such as notes, diagrams, or scaling considerations, are encouraged.
- The recommended time to complete the task is **5 hours**. This is intended to reflect the expected level of complexity.