**Back-end Design Architecture**

**1. Architecture**

CEP REST API

HR Portal REST API

Microservice 1

Microservice 2

Authentication

Authorization

Security

API Gateway

User

User

User

Database Service

Service Registry

WEB UI

**Fig. 1.1:** **CEP + HR back end micro service architecture high level design**

**Features:**

1. Users of the system will use **Web UI** to access the software.
2. Web UI will access the services via the **API Gateway**.
3. API Gateway is a middleware software that is responsible for **Security, Authentication, and Authorization**.
4. **Security** deals with the security of the application. E.g. the server certificates
5. **Authentication** deals with the authenticating the users of the system who will be trying to access the system.
6. **Authorization** will deal with the rights associated with every role. Since both the systems will be having different set of user roles, the service APIs will be relieved from the part of authorization. The Service API will purely focus on the fulfilling the tasks allocated to each of them.
7. Each **Microservice** will be a monolith in its own. This will help in having a coupling between the individual application.
8. Each microservice will be able to scale independently. Communication between the services will be restricted to sharing common details required for both the services. Any computation involved will not be shared. This might increase the load, but it will help in maintaining the state of the application at any given instance of time.
9. If a new application is to be plugged in, then the new application will have it’s own set of microservices. This will decouple different applications but will bind each individual application to its own service.
10. Individual microservices will communicate with the database independently thus reducing the load time required to load an application.

**2. Technology Stack**

1. User will interact with the UI using different set of devices(e.g. Mobile, Desktop, etc.)
2. Web UI will be based on Angular 6+. Web UI will be communicating with the API Gateway for service consumption.
3. API Gateway will be based on Node.js
4. Service registry will be an isolated component connected with API Gateway and will be using Node.js
5. CEP Rest API will use Node.js for service API
6. HR Portal will use Python for service API
7. Database used for CEP and HR Portal will be using MySQL/PostgreSQL as database.

**3. CEP Rest API:**

1. This will be a monolithic architecture that will be responsible for catering the needs for the CEP application.
2. A set of APIs that will be responsible for entire communication for the CEP portal and database.
3. Database used will be MySQL or PostgreSQL.
4. Since our applications will be entirely distinguished entities, we will be having a monolithic architecture for individual entities. This will assist in the deploying of the application independently.
5. **HR Portal Rest API:**
6. This will be a monolith as well catering the requirements for HR portal.
7. Technology used will be python as back-end.
8. Database used will be mongo db for adjusting the ever changing demands of the HR portal.