**High-Level Design (HLD) – Task Service**

**1. Overview**

The Task Service allows users to create, assign, and track tasks. It integrates with **Apache Camel** for routing and **Kafka** for event-driven communication with other services (e.g., Notification Service).

**2. Architecture**

* **Frontend**: React / Angular interface for user interaction
* **Backend**: REST API (Spring Boot / Node.js / Django)
* **Database**: PostgreSQL / MongoDB
* **Message Broker**: Kafka
* **Integration**: Apache Camel routes tasks to Kafka

**3. Modules**

* **Task Management**
  + Create / Update / Delete Task
  + Assign Task
  + Status Tracking (Open, In Progress, Completed)
* **Kafka Integration**
  + Produce task events (e.g., task.created)
* **Apache Camel**
  + Middleware to transform and route messages
* **Notification Trigger**
  + Notify user via Notification Service (Kafka Consumer)

**4. Data Flow**

1. User sends a task creation request
2. Backend stores task in DB
3. Apache Camel picks and transforms the message
4. Sends to Kafka
5. Notification Service consumes and notifies user

**5. Database Schema (Simplified)**

CREATE TABLE tasks (

id UUID PRIMARY KEY,

title VARCHAR(100),

description TEXT,

assigned\_to UUID,

status VARCHAR(20),

due\_date DATE,

created\_at TIMESTAMP,

updated\_at TIMESTAMP

);

**6. Dependencies**

* Apache Kafka
* Apache Camel
* PostgreSQL/MongoDB
* Notification Service

**7. Security**

* JWT-based Auth
* Role-based Authorization

**8. Scalability**

* Kafka allows async processing
* Camel decouples task generation from consumers