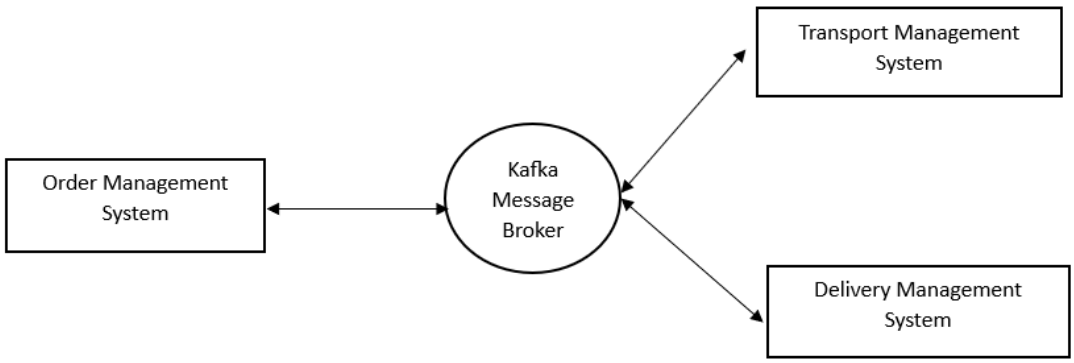
System Architecture

System Architecture for OMS, TMS, and DMS

Order Management System (OMS), Transportation Management System (TMS), and Distribution Management System (DMS), here's a detailed system architecture that shows the interactions, communication, and overall structure of the unified system. The architecture leverages **Kafka** for communication between the different services.



1. OMS (Order Management System)

Responsibilities:

- Capture and validate customer orders.
- Handle customer and order data.
- Interact with **TMS** for transportation and shipment creation.

Components:

- **REST API (Controller)**:
    - Handles incoming HTTP requests for placing orders, viewing order status, and retrieving order information.
- **Service Layer**:
    - Business logic for handling order placement, including saving orders to the database.
    - Sends order events to Kafka topic for TMS and DMS to process.

- **Repository Layer**:
    - Manages data persistence using JPA for the Orders, Customers, and OrderItems entities.
- **Kafka Producer**:
    - Sends messages to Kafka topic order-events when a new order is placed.

2. TMS (Transportation Management System)

Responsibilities:

- Route planning and optimization.
- Carrier management and assignment.
- Freight auditing and shipment tracking.

Components:

- **REST API (Controller)**:
    - Exposes endpoints for managing shipments, assigning carriers, and retrieving real-time tracking.
- **Service Layer**:
    - Listens to Kafka events from the OMS (order-events) for new orders.
    - Processes route optimization based on distance and delivery location.
    - Manages carrier assignment and shipment creation.
    - Sends shipment details to Kafka topic shipment-events for DMS to process.
- **Repository Layer**:
    - Manages data persistence for Shipments, Routes, and Carriers entities.

- **Kafka Consumer**:
    - Listens to Kafka topic order-events for new orders.
- **Kafka Producer**:
    - Sends shipment-related events to Kafka topic shipment-events.

3. DMS (Distribution Management System)

Responsibilities:

- Warehouse management (inventory control, stock levels).
- Handle product returns and restocking.
- Real-time tracking of stock and shipments across warehouses.

Components:

- **REST API (Controller)**:
    - Provides endpoints for managing warehouse data, inventory updates, and product returns.
- **Service Layer**:
    - Listens to Kafka events from TMS (shipment-events) for new shipments.
    - Updates inventory levels in the warehouse.
    - Handles product returns and restocking.
    - Uses warehouse automation to optimize inventory levels and manage stock.
- **Repository Layer**:
    - Manages data persistence for Warehouses, Inventory, and ProductReturn.
- **Kafka Consumer**:
    - Listens to Kafka topic shipment-events for new shipments to update the warehouse system.

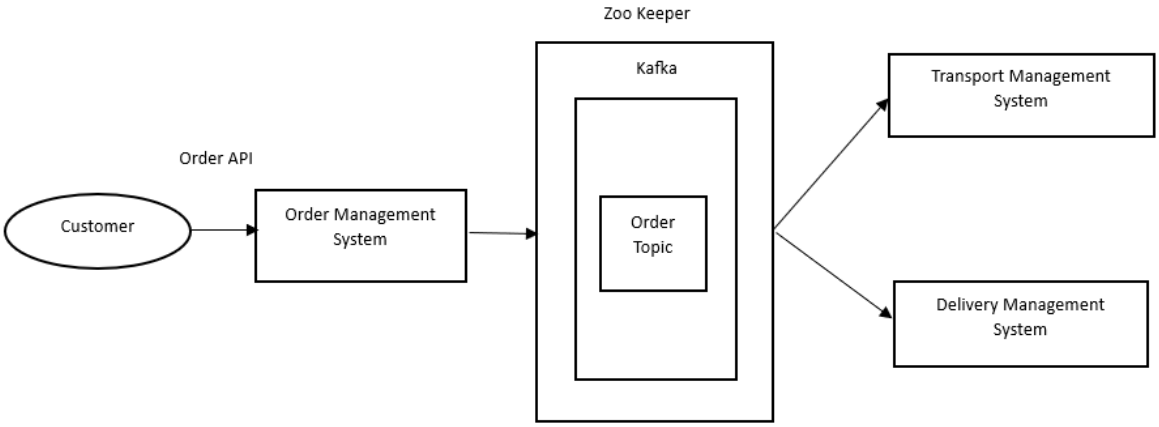
Overall System Flow

1. **Order Creation (OMS)**:
    - A customer places an order via the **OMS API**.
    - OMS validates and saves the order, then produces a Kafka message to the order-events topic.
2. **Order Processing (TMS)**:
    - **TMS Kafka Consumer** listens to order-events and retrieves the new order.
    - TMS optimizes the route, assigns a carrier, and creates a shipment.
    - TMS then produces a Kafka message to the shipment-events topic containing shipment details.
3. **Shipment and Inventory Updates (DMS)**:
    - **DMS Kafka Consumer** listens to shipment-events and processes the shipment for inventory control.
    - DMS updates stock levels, tracks the shipment, and handles product returns if necessary.

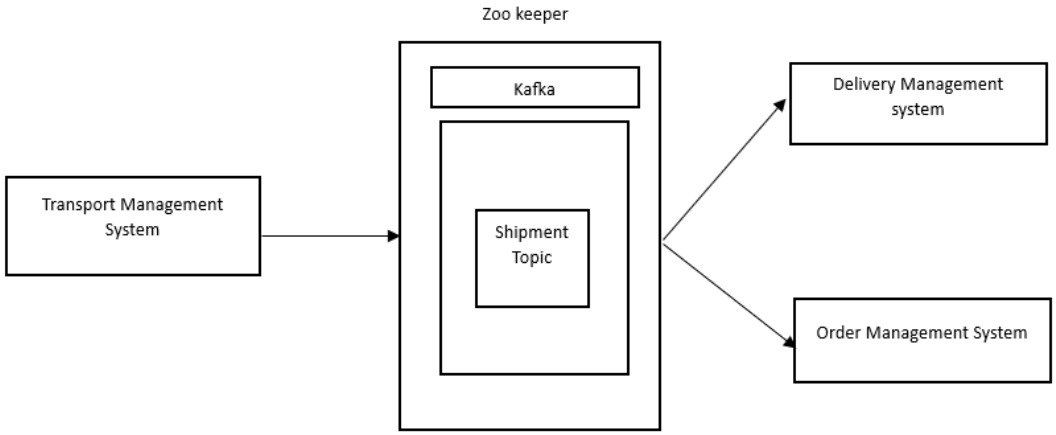
Architecture Diagram Breakdown

High-Level Architecture

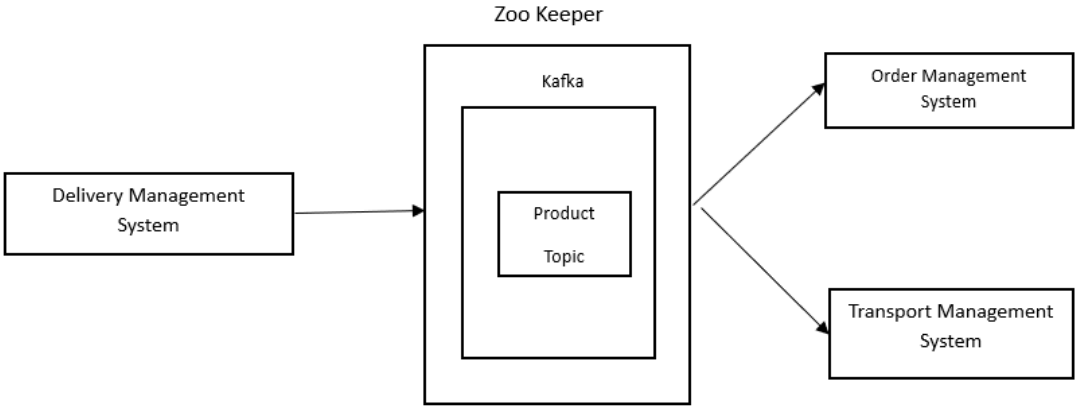
- **Frontend Clients (e.g., Web or Mobile)**: Interact with OMS APIs to place orders.
- **Order Management System (OMS)**:
    - REST API (Order Creation, Order Status)
    - Kafka Producer to send orders to order-events.
    - MySQL Database for orders and customer data.



- **Transportation Management System (TMS)**:
  - REST API (Shipment Management, Carrier Assignment)
  - Kafka Consumer for order-events.
  - Route Optimization Engine.
  - Kafka Producer to send shipments to shipment-events.
  - MySQL Database for shipments, carriers, routes.



- **Distribution Management System (DMS)**:
  - REST API (Warehouse Management, Inventory, Product Returns)
  - Kafka Consumer for shipment-events.
  - Kafka Producer to send shipments to shipment-events.
  - MySQL Database for inventory, warehouses, returns.



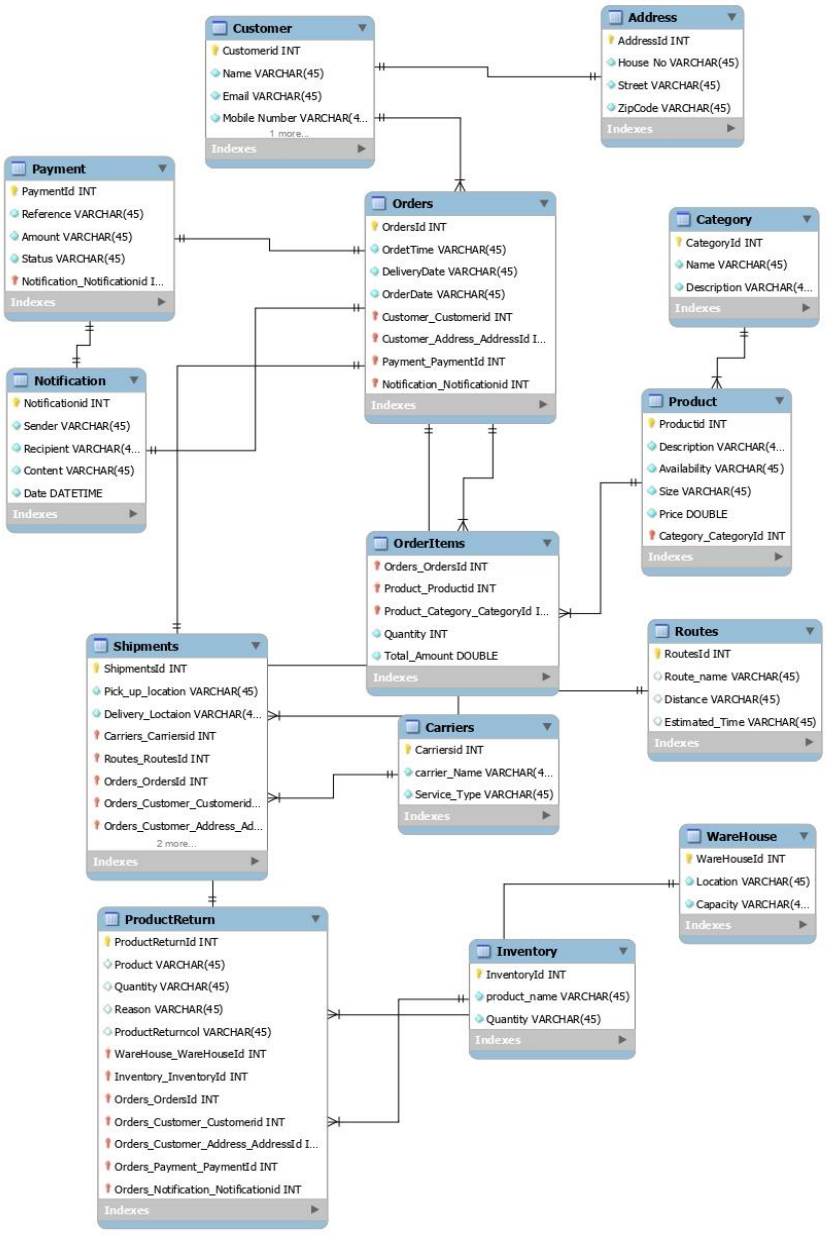
Detailed Module Interaction

1. **Order Management System (OMS)**:
    - Responsible for handling customer orders.
    - Communicates with TMS via Kafka to create shipments.
2. **Transportation Management System (TMS)**:
    - Listens for new orders from OMS, optimizes routes, assigns carriers.
    - Updates DMS about shipment statuses via Kafka.
3. **Distribution Management System (DMS)**:
    - Listens for shipment events from TMS and updates inventory accordingly.
    - Manages returns and ensures accurate stock levels.

Conclusion

This architecture shows how each module is decoupled, allowing them to scale and evolve independently. Kafka ensures reliable, asynchronous communication between OMS, TMS, and DMS, while Spring Boot handles the REST APIs, data persistence, and business logic. Each service is responsible for specific domains in the supply chain management process.

ER Diagram:


Customer
Customerid INT
Name VARCHAR(45)
Email VARCHAR(45)
Mobile Number VARCHAR(4...
1 more...
Indexes
Address
AddressId INT
House No VARCHAR(45)
Street VARCHAR(45)
ZipCode VARCHAR(45)
Indexes
Payment
PaymentId INT
Reference VARCHAR(45)
Amount VARCHAR(45)
Status VARCHAR(45)
Notification_Notificationid I...
Indexes
Orders
OrdersId INT
OrdetTime VARCHAR(45)
DeliveryDate VARCHAR(45)
OrderDate VARCHAR(45)
Customer_Customerid INT
Customer_Address_AddressId I...
Payment_PaymentId INT
Notification_Notificationid INT
Indexes
Category
CategoryId INT
Name VARCHAR(45)
Description VARCHAR(4...
Indexes
Notification
Notificationid INT
Sender VARCHAR(45)
Recipient VARCHAR(4...
Content VARCHAR(45)
Date DATETIME
Indexes
Product
Productid INT
Description VARCHAR(4...
Availability VARCHAR(45)
Size VARCHAR(45)
Price DOUBLE
Category_CategoryId INT
Indexes
OrderItems
Orders_OrdersId INT
Product_Productid INT
Product_Category_CategoryId I...
Quantity INT
Total_Amount DOUBLE
Indexes
Shipments
ShipmentsId INT
Pick_up_location VARCHAR(45)
Delivery_Loctaion VARCHAR(4...
Carriers_Carriersid INT
Routes_RoutesId INT
Orders_OrdersId INT
Orders_Customer_Customerid...
Orders_Customer_Address_Ad...
2 more...
Indexes
Carriers
Carriersid INT
carrier_Name VARCHAR(4...
Service_Type VARCHAR(45)
Indexes
Routes
RoutesId INT
Route_name VARCHAR(45)
Distance VARCHAR(45)
Estimated_Time VARCHAR(45)
Indexes
WareHouse
WareHouseId INT
Location VARCHAR(45)
Capacity VARCHAR(4...
Indexes
ProductReturn
ProductReturnId INT
Product VARCHAR(45)
Quantity VARCHAR(45)
Reason VARCHAR(45)
ProductReturncol VARCHAR(45)
WareHouse_WareHouseId INT
Inventory_InventoryId INT
Orders_OrdersId INT
Orders_Customer_Customerid INT
Orders_Customer_Address_AddressId I...
Orders_Payment_PaymentId INT
Orders_Notification_Notificationid INT
Indexes
Inventory
InventoryId INT
product_name VARCHAR(45)
Quantity VARCHAR(45)
Indexes