1. Kafka Producers

Here's how to implement the producers with message filtering:

Producer 1: Inventory Orders Producer

1. **Message Filtering**:
    - While receiving a message, check if the message contains a field named "type" (case-sensitive).
    - If the "type" field exists and its value is "inventory", proceed to send the message.
2. **Sending Message**:
    - Use the Kafka client library to create a Kafka producer instance with necessary configuration.
    - Define the target Kafka topic where the filtered messages will be sent.
    - For each filtered message, convert it into a Kafka record (key-value pair, key can be optional).
    - Send the record using the producer instance.

Producer 2: Delivery Orders Producer

Follow the same steps as Producer 1, but filter messages where the "type" field value is "delivery".

Choosing Key (Optional):

- You can consider using a unique identifier from the message (e.g., order ID) as the key for better record organization within the topic.

2. Kafka Consumers

Consumer 1: Inventory Data Consumer

1. **Topic Subscription**:
    - Use the Kafka client library to create a Kafka consumer instance with necessary configuration (e.g., group ID).
    - Subscribe the consumer to the Kafka topic where inventory messages are being sent.
2. **Message Processing**:
    - Implement a callback function that gets triggered whenever the consumer receives a message.
    - Within the callback:
        - Check if the message contains a "type" field.
        - If the "type" field exists and its value is "inventory", process the message data (e.g., update inventory database).

Consumer 2: Delivery Data Consumer

Follow the same steps as Consumer 1, but process messages where the "type" field value is "delivery" (perform delivery related actions).

Error Handling:

- Implement proper error handling mechanisms for both producers and consumers to handle potential exceptions during message sending/receiving.

3. Message Filtering Logic

Implementation of the filtering logic will depend on your chosen programming language.

Here's a general approach:

1. Define a function that takes a message as input.
2. Check if the message contains a field named "type" (case-sensitive).
3. If the "type" field exists, compare its value with the desired type (e.g., "inventory" or "delivery").
4. If the type matches, return True indicating the message needs to be sent.
5. Otherwise, return False to discard the message.

This ensures only messages with the desired "type" field value are sent to the Kafka topic by the producer.