Event-driven Architecture

Systems Analysis & Design

Learning Objectives

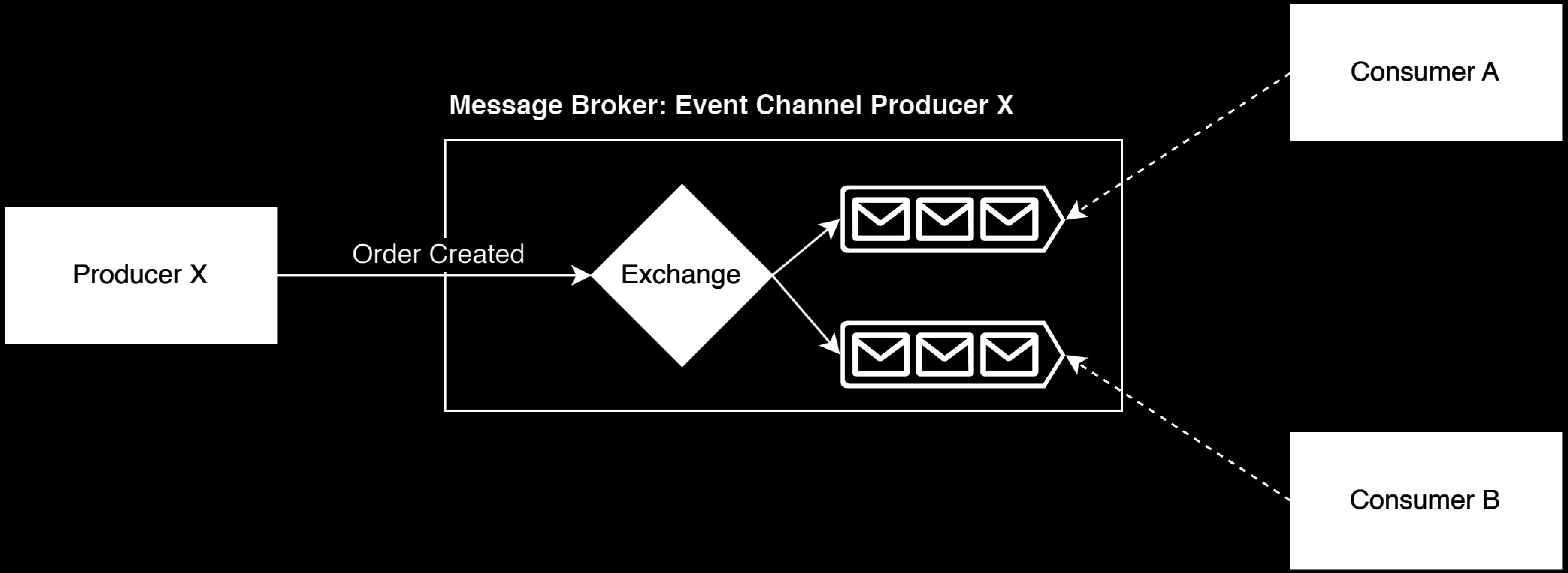
By the end of this session, you will have acquired the following information:

- Command vs Event
- Event-driven Architecture
- Self-contained Service
- Orchestration Pattern
- Choreography Pattern

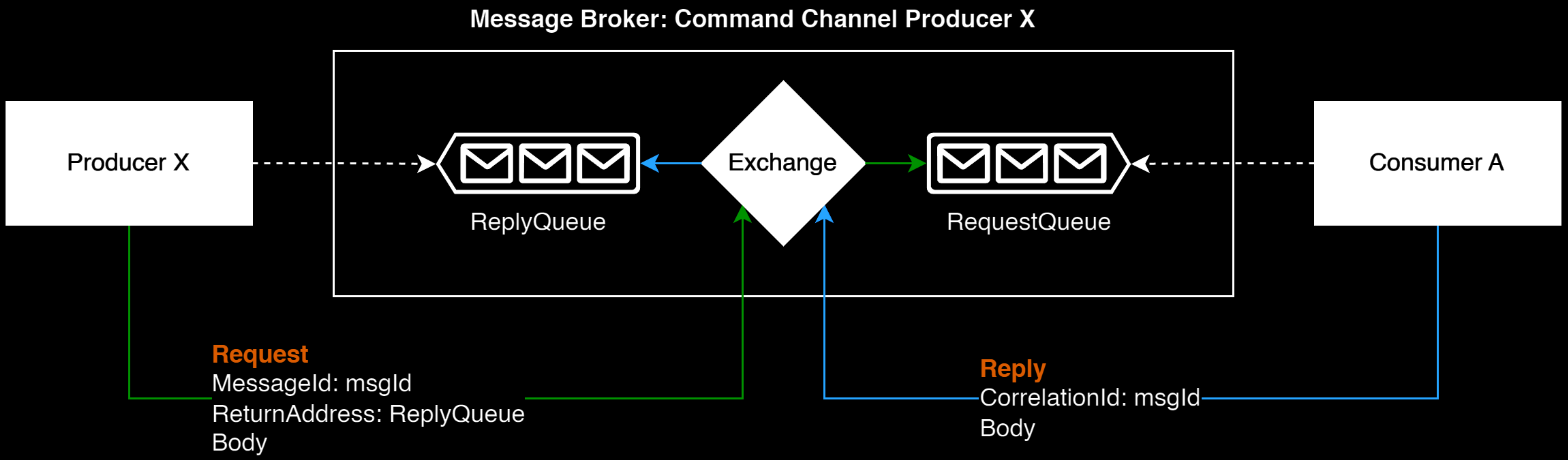
Command vs Event

- **Command**
  - A message that is the equivalent of a request. It specifies the operation to invoke and its parameters.
- **Event**
  - A message indicating that something notable has occurred in the sender. An event is often a domain event, which represents a state change of a domain object such as an `Order`, or a `Customer`.

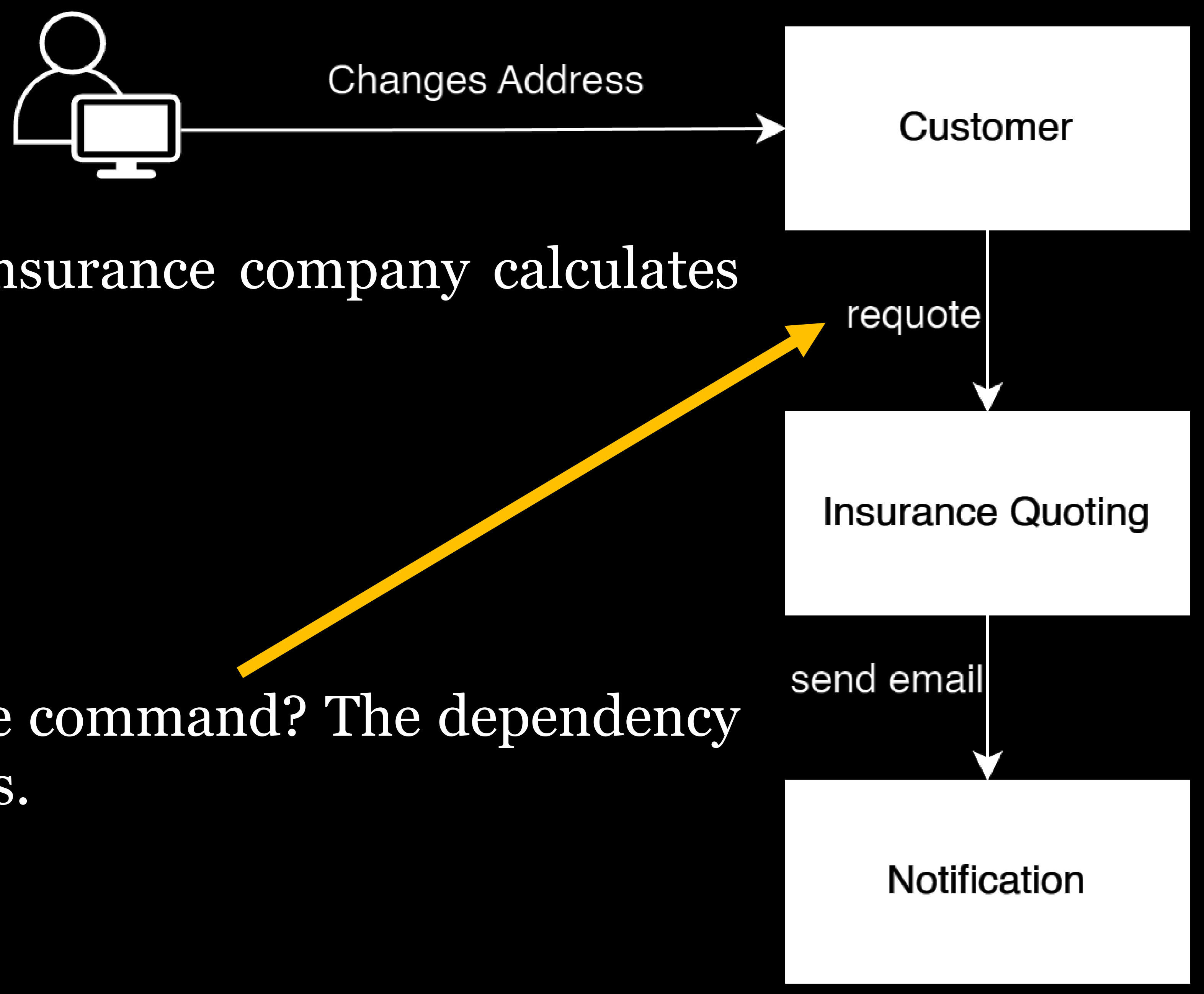
Event Channel



Command Channel



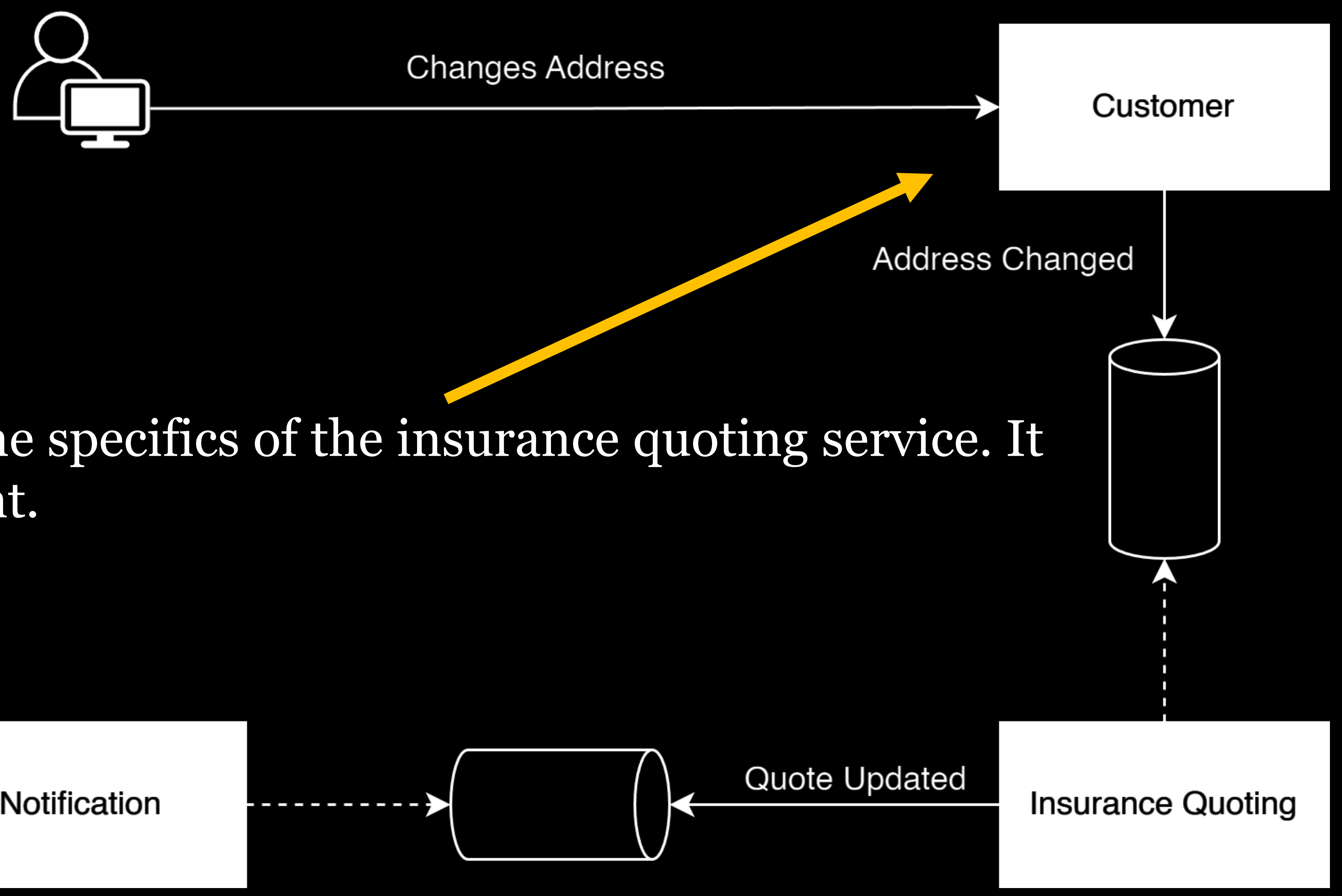
Illustrative Example



Insurance quoting is the process by which an insurance company calculates the cost of providing insurance for something.

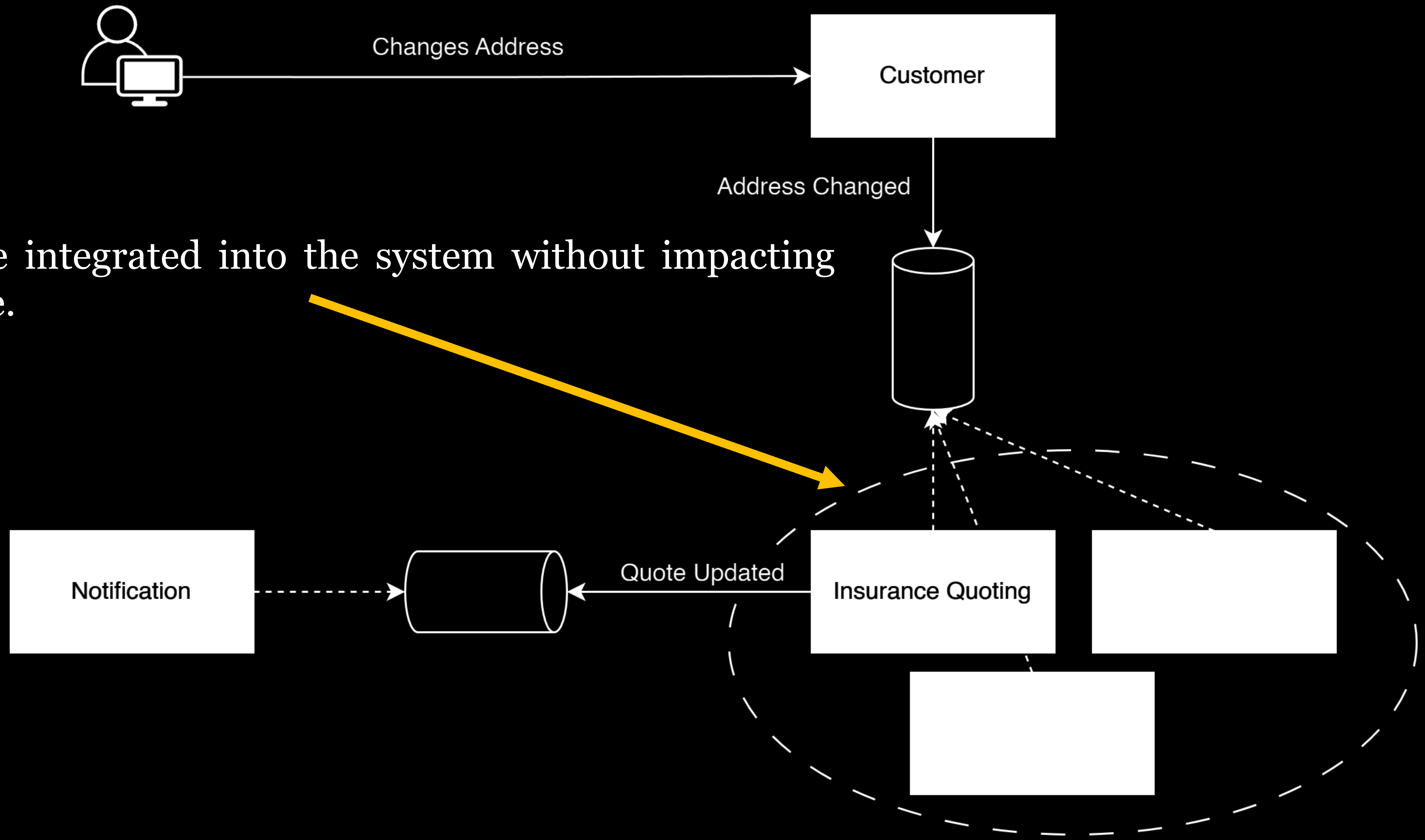
Why should customer service handle the requote command? The dependency should be reversed to reflect real-world processes.

Event-driven Architecture

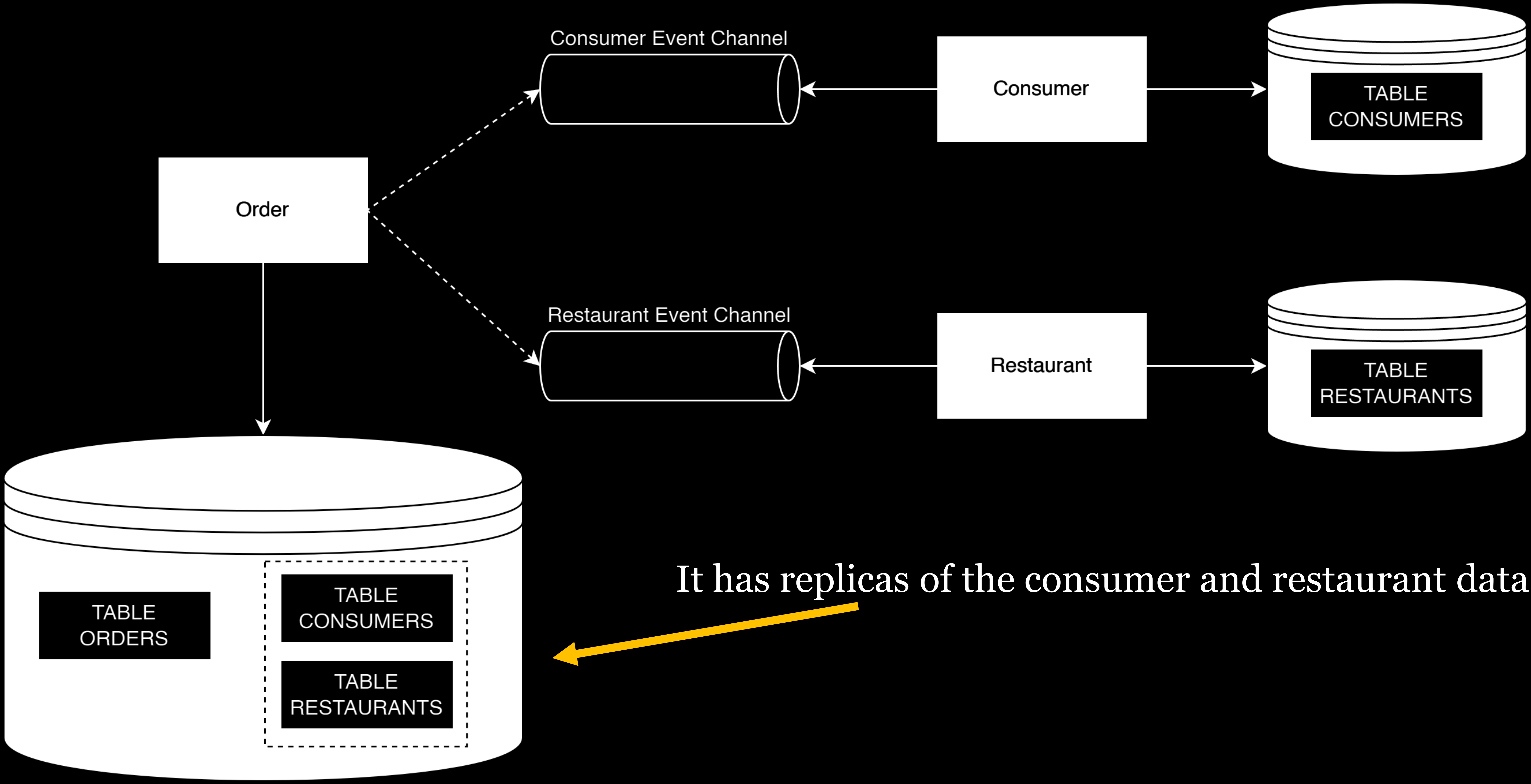


The Customer service isn't aware of the specifics of the insurance quoting service. It publishes an `Address Changed` event.

New services can be integrated into the system without impacting the Customer service.



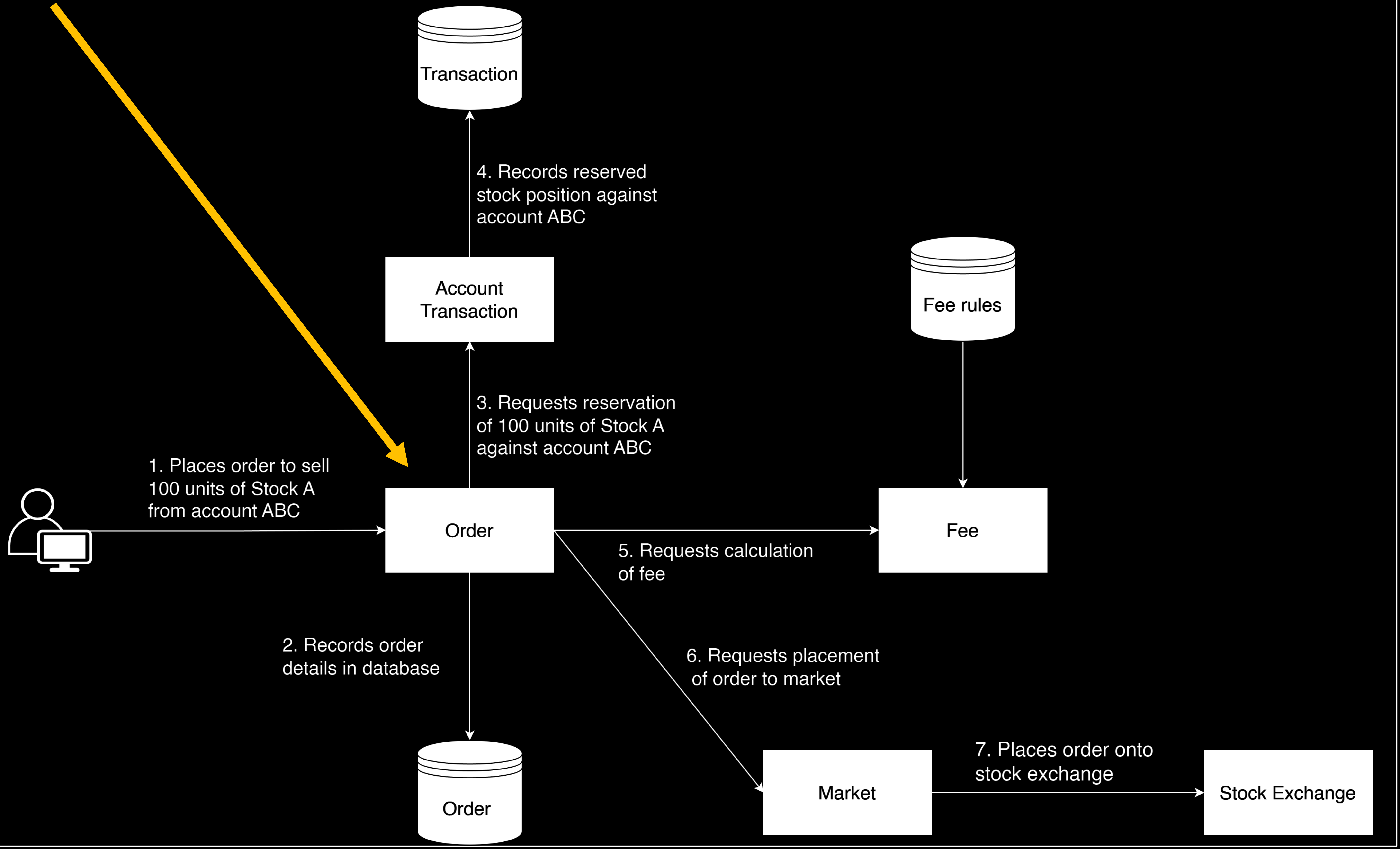
Self-contained Service



It has replicas of the consumer and restaurant data.

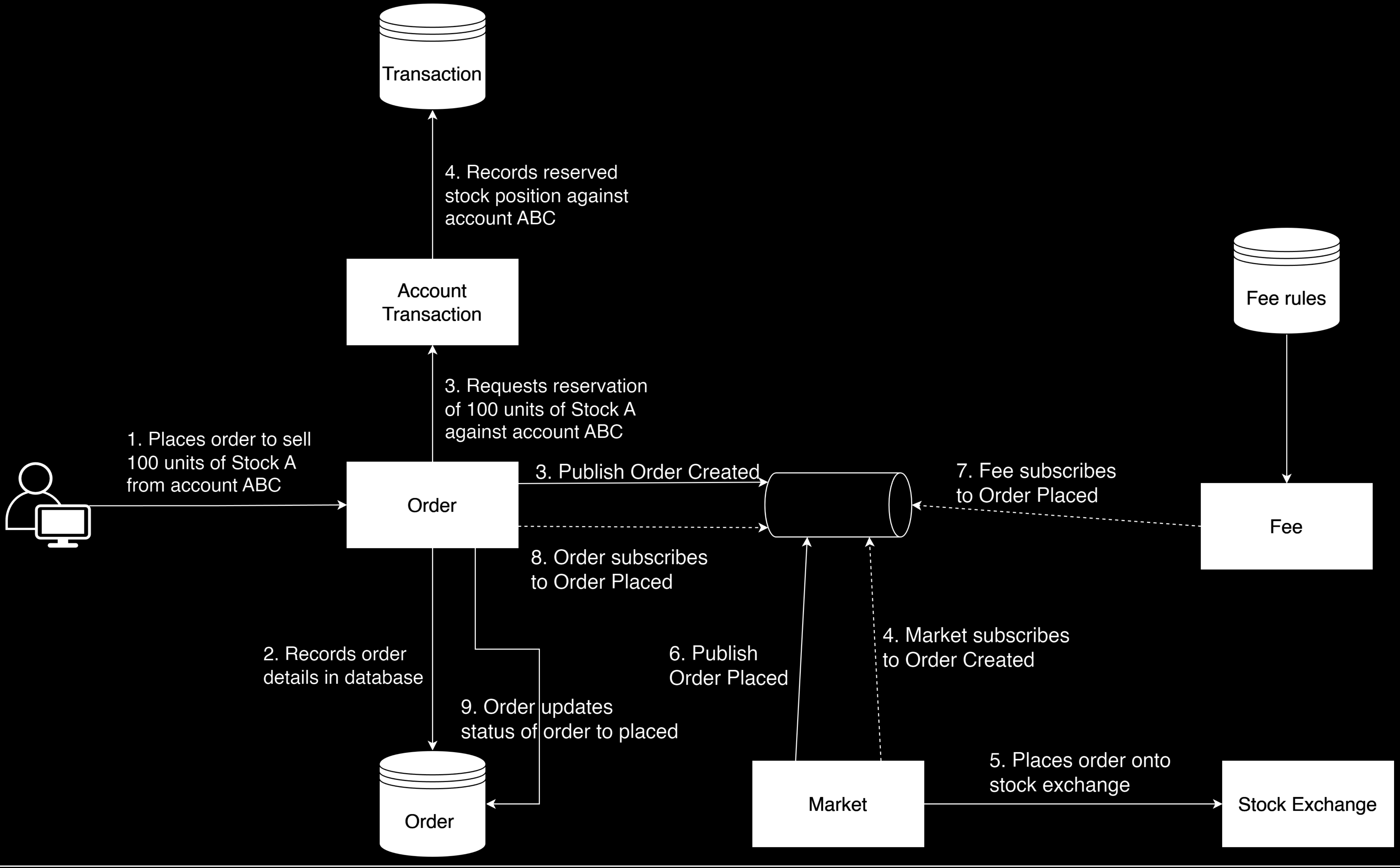
Orchestration Pattern

The orchestrator manages complex business flows by calling independently deployed services, handling exceptions, retrying requests, maintaining state, and returning the final response.


Transaction
4. Records reserved stock position against account ABC
Account Transaction
Fee rules
3. Requests reservation of 100 units of Stock A against account ABC
1. Places order to sell 100 units of Stock A from account ABC
Order
Fee
5. Requests calculation of fee
2. Records order details in database
6. Requests placement of order to market
7. Places order onto stock exchange
Market
Stock Exchange
Order

Choreography Pattern

Each service participates in the decision-making process about the workflow of a business transaction, instead of relying on a central point of control.


Transaction
4. Records reserved stock position against account ABC
Account Transaction
Fee rules
3. Requests reservation of 100 units of Stock A against account ABC
1. Places order to sell 100 units of Stock A from account ABC
Order
3. Publish Order Created
7. Fee subscribes to Order Placed
Fee
8. Order subscribes to Order Placed
4. Market subscribes to Order Created
6. Publish Order Placed
2. Records order details in database
9. Order updates status of order to placed
5. Places order onto stock exchange
Market
Stock Exchange
Order

Further Resources

- Inter-Process Communication in a Microservice Architecture