

Message Queue - Discussion Topics

Architecture & Design

1. When would you choose Kafka over RabbitMQ or SQS?

- Event streaming vs task queues
- Replay requirements
- Throughput considerations

2. How do you decide between choreography and orchestration?

- Workflow complexity
- Visibility requirements
- Team boundaries

3. What are the trade-offs of event sourcing?

- Storage growth
- Query complexity
- Event schema evolution

Real-World Scenarios

1. Design an order processing pipeline for an e-commerce platform

- Service boundaries
- Failure handling
- Consistency guarantees

2. How would you handle a consumer that can't keep up with message volume?

- Scaling strategies
- Backpressure mechanisms
- Priority queuing

3. Your payment service is down - how do you handle pending orders?

- Saga compensation
- Retry strategies
- Dead letter queue handling

Performance & Reliability

1. How do you ensure exactly-once message processing?

- Idempotency patterns
- Deduplication strategies
- Transaction support

2. What's your approach to message schema evolution?

- Backward compatibility
- Versioning strategies
- Consumer migration

Advanced Topics

1. When would you use CQRS in your architecture?

- Read/write ratio considerations
- Consistency requirements
- Complexity trade-offs

2. How do you debug issues in an event-driven system?

- Distributed tracing
- Correlation IDs
- Event replay for debugging