



CLOUD COMPUTING APPLICATIONS

Decoupling in Cloud Architectures

Prof. Reza Farivar

Multi-tier Distributed Architecture

- Enterprise architectures require elasticity and scalability
 - Scalability: respond to increasing demand
 - Elasticity: respond to decreasing demand
- Fault tolerance
 - Component failure is the rule in cloud environments
- Changing demand patterns
 - Hard to predict how many resources we will need in the future
- Complexity
 - Multiple platforms and development teams

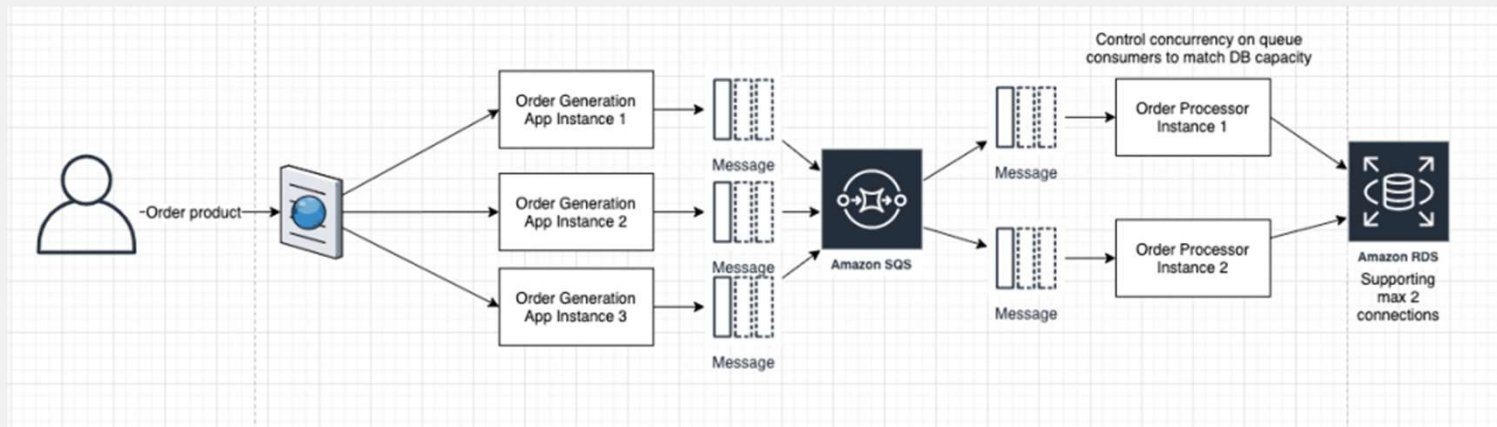
Decoupling

- The key to achieving reliable, scalable, elastic architectures is decoupling
- Building applications from individual components that each perform a discrete function
- A reliable queue between components
 - Allows many integration patterns for connecting services
- Loose coupling → increases an architecture's resiliency to failure and ability to handle traffic spikes
 - producer and consumer operate independently
- Asynchronous Communication

Message Queues

- Message queues → decouple and scale microservices, distributed systems, and serverless applications
 - Send, store, and receive messages between software components
 - Any volume
 - Without losing messages
 - No need to rely on other services be always available
- Can easily handle momentary spikes in demands
 - Up spikes and down spikes
- Guaranteed message delivery
 - At least Once
 - Exactly Once

Example



<https://aws.amazon.com/blogs/architecture/application-integration-using-queues-and-messages/>

Message Queue Platforms

- Open Source
 - Apache ActiveMQ
 - Apache RabbitMQ
 - Apache Kafka
- Proprietary / cloud services
 - Amazon AWS Simple Queue Service
 - Amazon MQ
 - Apache ActiveMQ
 - Apache RabbitMQ
 - Amazon Kinesis
 - Amazon Managed Kafka

Publish-Subscribe Model

- A sibling of the message queue systems
- Producers publish messages to the queue
- Several consumers, having subscribed to a specific producer (or topic, etc.), all receive the message
- Publishers and subscribers are decoupled
- Example:
 - Kafka
 - AWS Simple Notification Service