

Messaging Systems with Python

Understand Messaging Systems and Event-driven Architectures



Shubham Sinha

Software Engineer | Servicenow

linkedin@shubham-sinha-psight



Course Introduction

Introduction to messaging systems

Message queues vs event streams

Real world implementation

**How to scale data processing without
overloading systems**

- Synchronous processing bottlenecks
- Coupling in systems
- Data loss and failure recovery





Messaging System



Messaging System

Messaging systems enable communication between services or components by exchanging messages asynchronously



Purpose in Data Engineering

**Enables event-driven
architecture**

**System reliability and
fault tolerance**

Supports scalability



Components of a Messaging System

Producer

Message broker

Consumer

Topics/queues

**Message
acknowledgement
and persistence**





Message Queues and Event Streams



Message Queues vs. Event Streams

Message Queues

Task distribution

One-to-one message consumption

Short term storage, auto-deletion

Request-response, job queues

vs.

Event Stream

Continuous data stream

One-to-many broadcasting

Retains history for replay

Event driven architectures, log processing





Asynchronous Processing



Benefits of Asynchronous Processing

Decoupling

Scalability

Resilience

Performance





Messaging Systems vs. Direct Database Queries



Messaging Systems

**High throughput,
real-time processing
is required**

**Systems need to be
loosely coupled**

**Handling failures
with retry mechanism
is needed**



Direct Database Queries

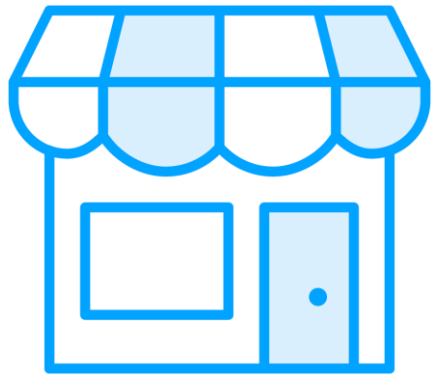
**Immediate,
consistent results are
needed**

**Data updates require
strong ACID
compliance**

**No event driven
behavior is necessary**



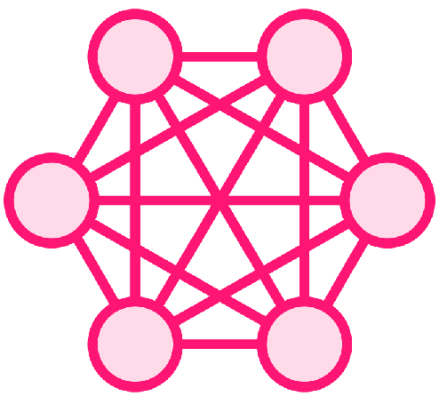
Real World Examples



Customer places an order



Payment processing



Shipping systems

