ONE PAGE SUMMARY

IT & Software › Microservices › Event Driven Architecture

Introduction to Event Driven Architecture

Unlocking the Power of Event Driven Architecture in Microservices

DEFINITION

> Event Driven Architecture (EDA) is a software design pattern where events trigger actions in systems. It allows applications to respond in real-time to changes and events.

KEY CONCEPTS

Events

An event is a significant change in state that can trigger actions within the system.

Event Producers

These are components that generate events when a change occurs, such as user actions or system updates.

Event Consumers

These are components that listen for and respond to events, executing specific actions based on them.

Message Brokers

Message brokers facilitate communication between event producers and consumers, ensuring reliable message delivery.

Loose Coupling

EDA promotes loose coupling between services, allowing them to operate independently and scale easily.

EXAMPLES

→ A user uploads a photo, triggering an event that processes the image.

→ An order is placed, generating an event that updates inventory and notifies shipping.

→ A sensor detects temperature change, sending an event to adjust the HVAC system.

MEMORY TIPS

★ Remember 'E' for Event: Everything starts with an event.

★ Think of producers as 'creators' and consumers as 'responders'.

★ Use the acronym 'MELT' for Message Brokers, Events, Loose Coupling, and Triggers.

COMMON MISTAKES

✗ Confusing events with commands; events describe what happened, commands request actions.

✗ Overloading event consumers with too many responsibilities.

✗ Neglecting the importance of message delivery guarantees in distributed systems.

QUICK RECAP

Event Driven Architecture enhances responsiveness and scalability in microservices. It relies on events to trigger actions, using producers, consumers, and message brokers to facilitate communication.