ONE PAGE SUMMARY

IT & Software › Software Architecture › Event Driven Design

# Introduction to Event Driven Design

Unlocking the Power of Event Driven Design in Software Architecture

## DEFINITION

> Event Driven Design is a software architecture paradigm where system components communicate through events. It allows for asynchronous processing and improves system scalability and responsiveness.

## KEY CONCEPTS

### Events

Events are significant changes in state or occurrences that trigger actions in a system.

### Event Producers

These are components that generate events when a specific action occurs, like a user clicking a button.

### Event Consumers

Event Consumers are components that listen for and respond to events, executing relevant actions.

### Event Bus

An Event Bus is a communication layer that facilitates the transmission of events between producers and consumers.

### Asynchronous Processing

This allows systems to handle events without blocking other operations, improving performance and user experience.

## EXAMPLES

→ A user registration system triggers a welcome email upon successful sign-up.

→ An online shopping cart updates inventory levels when an item is added or removed.

→ A social media app notifies users of new messages or likes in real-time.

## MEMORY TIPS

★ Think of events like ripples in a pond; one action creates multiple reactions.

★ Remember 'Producer-Consumer' as a supply chain: producers create, consumers use.

★ Use the acronym ECE: Events, Consumers, Events Bus to recall key elements.

## COMMON MISTAKES

✗ Confusing events with commands; events signify occurrences, while commands request actions.

✗ Neglecting to handle event failures, which can lead to data inconsistency.

✗ Overcomplicating event flows, making systems harder to maintain and understand.

## QUICK RECAP

Event Driven Design enhances system responsiveness by using events for communication. Key components include events, producers, consumers, and the event bus, enabling asynchronous processing for better scalability.