ONE PAGE SUMMARY

IT & Software › Microservices › Service Boundaries

Introduction to Service Boundaries

Understanding Service Boundaries in Microservices Architecture

DEFINITION

> Service boundaries define the limits of a microservice's responsibilities and interactions. They help in organizing services to ensure clear separation of concerns.

KEY CONCEPTS

Single Responsibility Principle

Each microservice should focus on a specific function or business capability.

Loose Coupling

Services should be independent, minimizing dependencies to enhance flexibility and scalability.

API Contracts

Defined interfaces that services use to communicate, ensuring consistent interaction.

Domain-Driven Design

A methodology that aligns service boundaries with business domains for better organization.

Data Ownership

Each service should own its data, preventing tight coupling through shared databases.

EXAMPLES

→ An e-commerce platform with separate services for user management and order processing.

→ A banking application with distinct services for account management and transaction processing.

MEMORY TIPS

★ Think 'one service, one job' to remember the Single Responsibility Principle.

★ Use 'API = Agreement' to recall the importance of API contracts.

★ Visualize a puzzle where each piece represents a service to understand boundaries.

COMMON MISTAKES

✗ Defining service boundaries too broadly, leading to monolithic behavior.

✗ Overlapping responsibilities between services, causing confusion and tight coupling.

✗ Neglecting to document API contracts, resulting in integration issues.

QUICK RECAP

Service boundaries are essential for defining the scope of microservices. They ensure services are independent and focused on specific tasks, promoting scalability and maintainability.