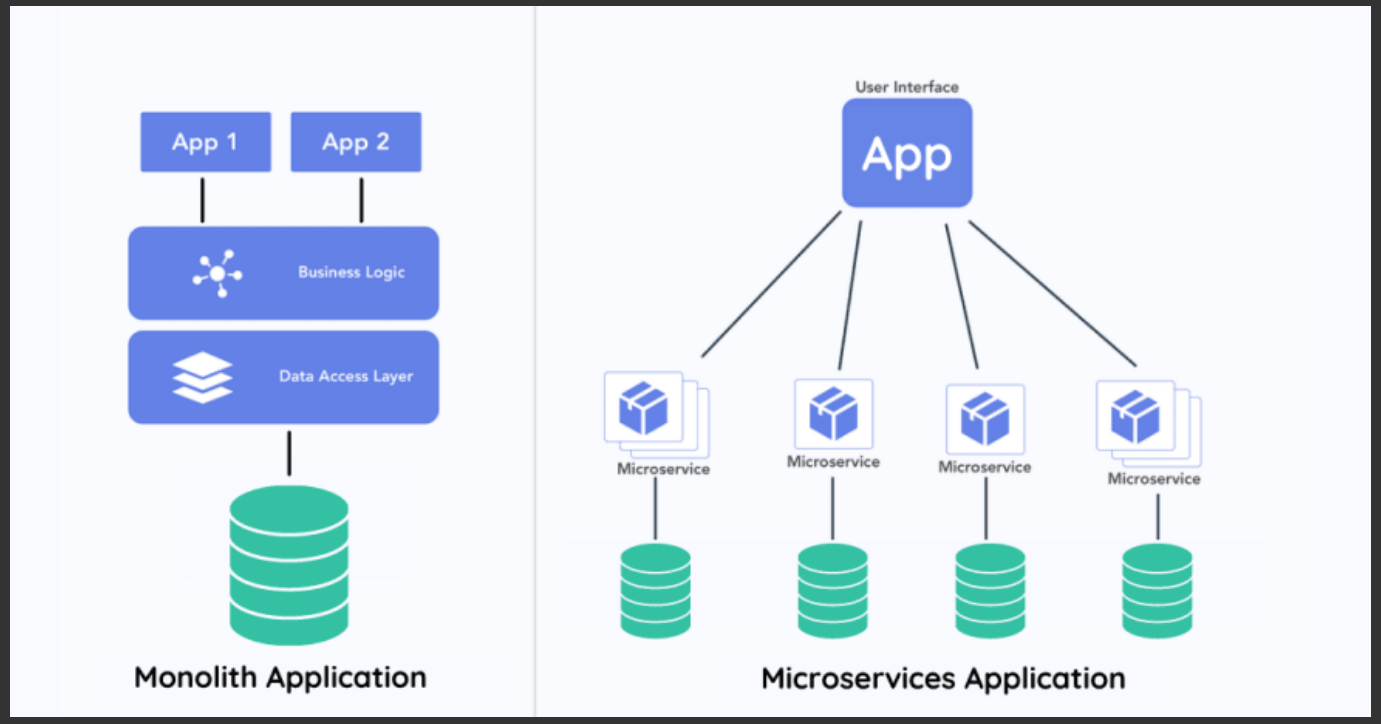
CODING SHUTTLE

Microservice

Introduction to Microservices

Microservices

Microservices are an architectural style where applications are developed as a collection of small, loosely coupled, independently deployable services.



Characteristics of Microservices

- Small, focused on doing one thing well
- Independently deployable
- Organized around business capabilities
- Decentralized data management
- Communication via lightweight protocols (typically HTTP/REST)

Comparison with Monolith

- Monolith: tightly coupled, single codebase
- Microservices: loosely coupled, multiple independent services

Challenges of Microservices

- Increased complexity in managing multiple services
- Distributed System Issues like Latency, load balancing, network reliability, and consistency
- Managing transactions and consistency across services
- Handling communication protocols (REST, gRPC, messaging)
- Monitoring and Logging needs centralized monitoring and logging solutions

Why use Microservices

- Scale services independently based on demand (e.g., scale only payment service during high traffic)
- Services can be written in different programming languages, suited to specific tasks
- Small teams can work on different services simultaneously, reducing time to market
- Failure in one service doesn't bring down the entire system
- Each service can be updated, deployed, and scaled independently
- Teams can work on different services without affecting each other