

Monolithic vs Microservices Architecture

Monolithic and Microservices are two widely used architectural styles in system design. Each approach solves different problems and comes with its own trade-offs.

1. What is Monolithic Architecture?

A monolithic architecture is a single, unified application where all components such as APIs, business logic, and database access are tightly integrated and deployed together.

- 1 Single codebase and single deployment unit
- 2 Usually backed by a shared database
- 3 Simple to build, test, and deploy initially
- 4 Scaling requires scaling the entire application

2. What is Microservices Architecture?

Microservices architecture divides an application into smaller, independent services. Each service owns its logic, data, and deployment lifecycle.

- 1 Multiple independently deployable services
- 2 Database per service is preferred
- 3 Services communicate via APIs or event streams
- 4 Independent scaling and fault isolation

3. Key Differences

- 1 Monoliths are simpler but less flexible at scale
- 2 Microservices increase scalability but add operational complexity
- 3 Monolith failures can impact the whole system
- 4 Microservices failures are usually isolated

4. How to Choose the Right Architecture

Choose Monolithic when:

- 1 You are building an MVP or early-stage product
- 2 Team size is small
- 3 System requirements are still evolving

Choose Microservices when:

- 1 Application has high traffic and complex domains
- 2 Multiple teams need independent deployments

3 Scalability and resilience are critical

In practice, many large systems start as monoliths and evolve into microservices once scale and complexity demand it.