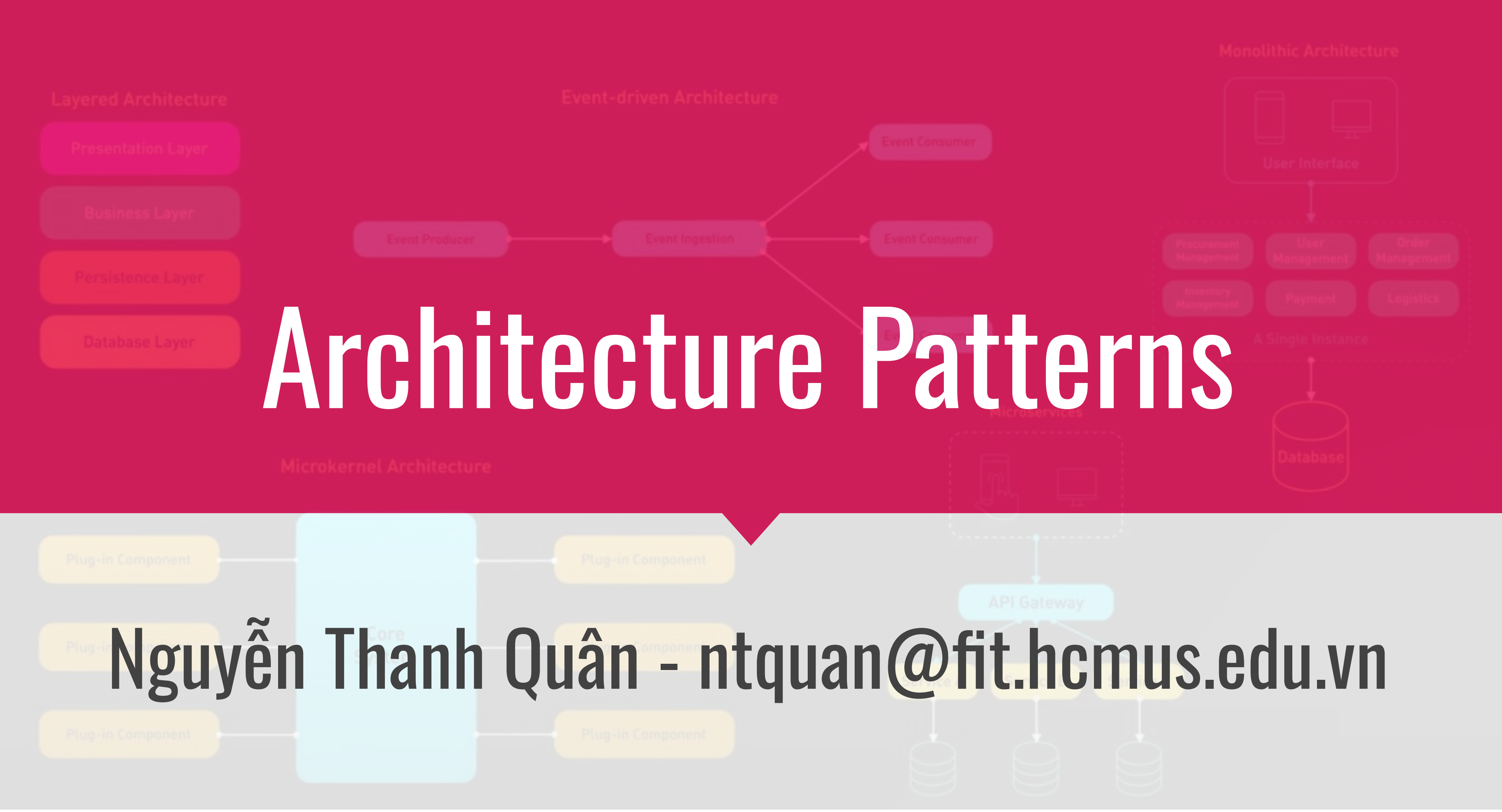


Architecture Patterns

Nguyễn Thanh Quân - ntquan@fit.hcmus.edu.vn

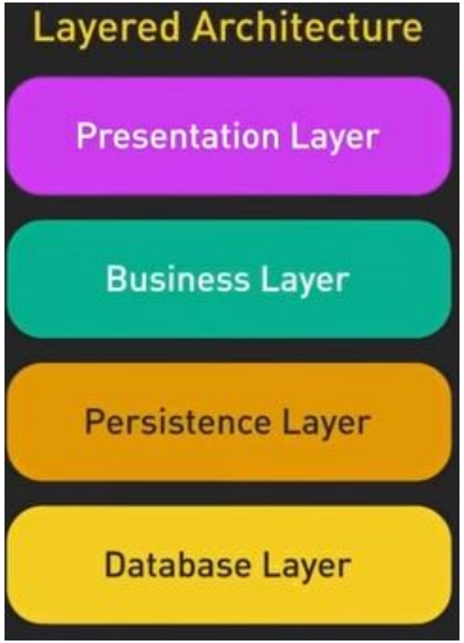
Agenda

1. Layered Architecture
2. Event-driven Architecture
3. Microkernel Architecture
4. Microservices Architecture
5. Monolithic Architecture
6. Serverless Architecture

Ref: https://www.youtube.com/watch?v=f6zXyq4VPP8

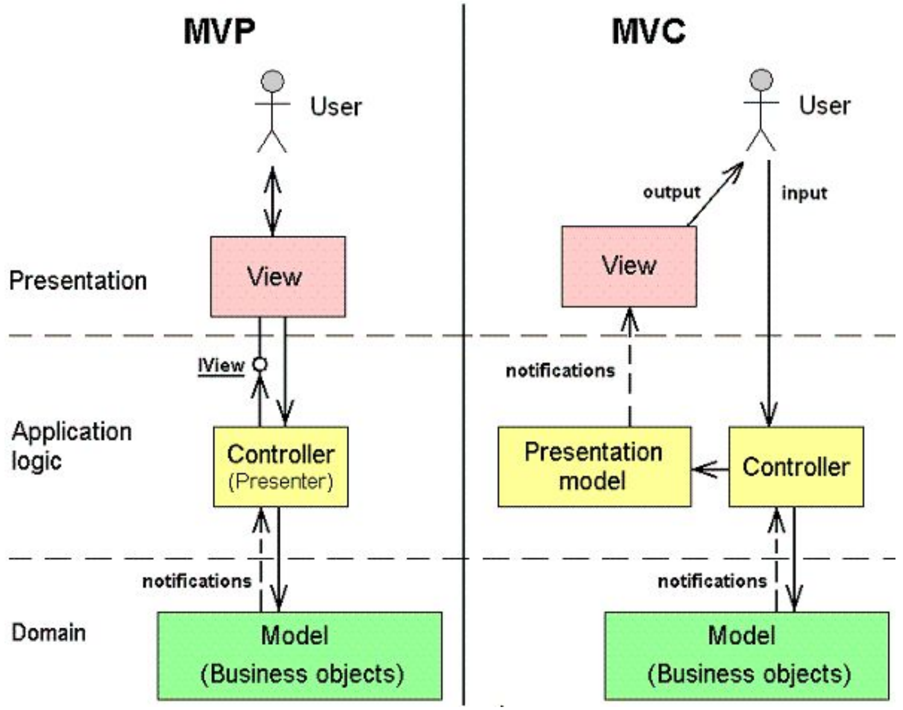
1. Layered Architecture

- Divides the application into multiple layers: Presentation (UI), Business Logic, Data Access.
- Pros: Easy maintenance, modular updates without affecting the entire system.
- Cons: Performance may be impacted due to multiple layers.
- Use cases: Enterprise applications, management software.



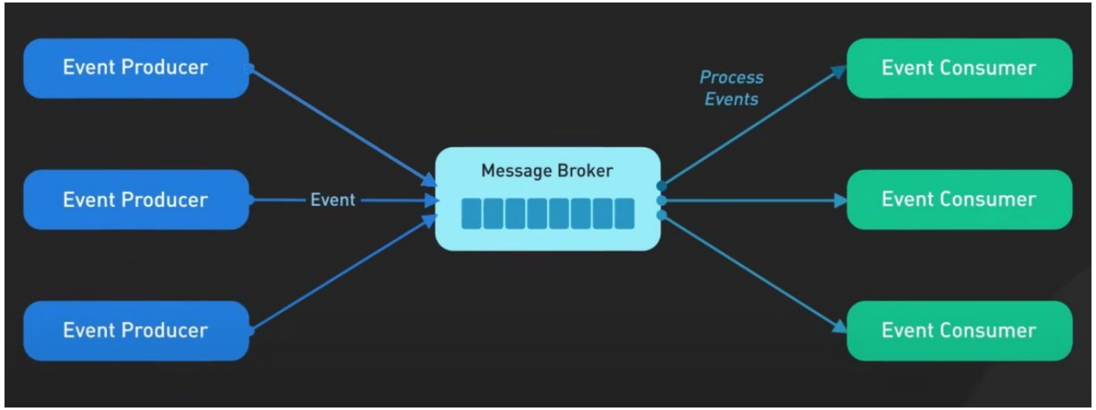
1. Layered Architecture

Web frameworks like Laravel, ASP.NET MVC, Django

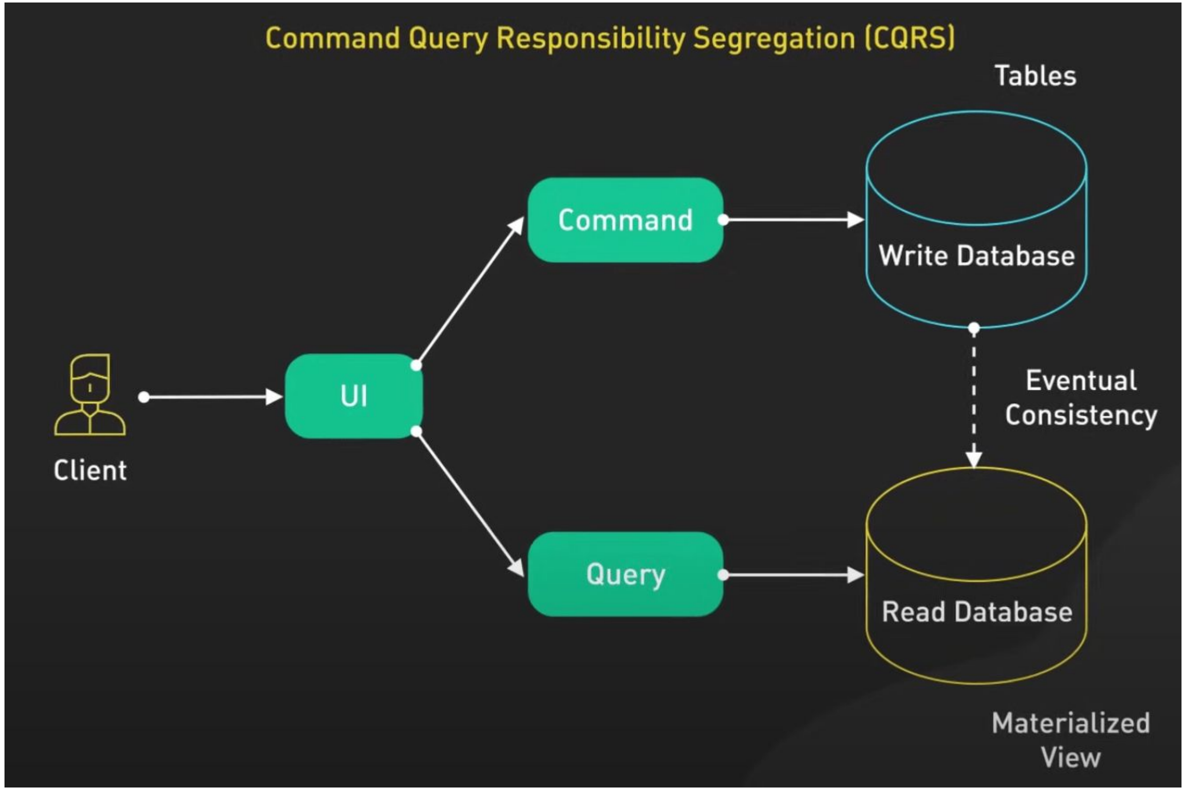


2. Event-driven Architecture

- Components communicate through events.
- Pros: Flexible, scalable, suitable for real-time processing.
- Cons: Debugging is challenging, requires strong monitoring.
- Use cases: IoT systems, streaming data processing (Kafka, RabbitMQ).

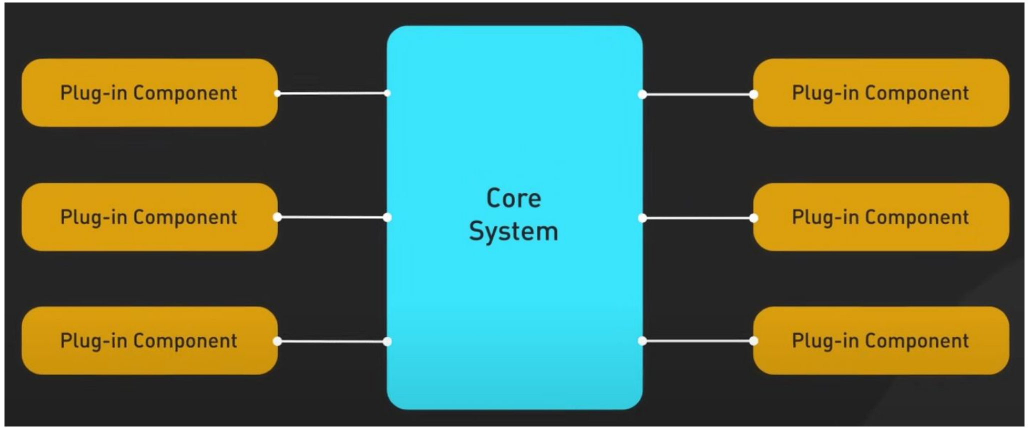


2. Event-driven Architecture

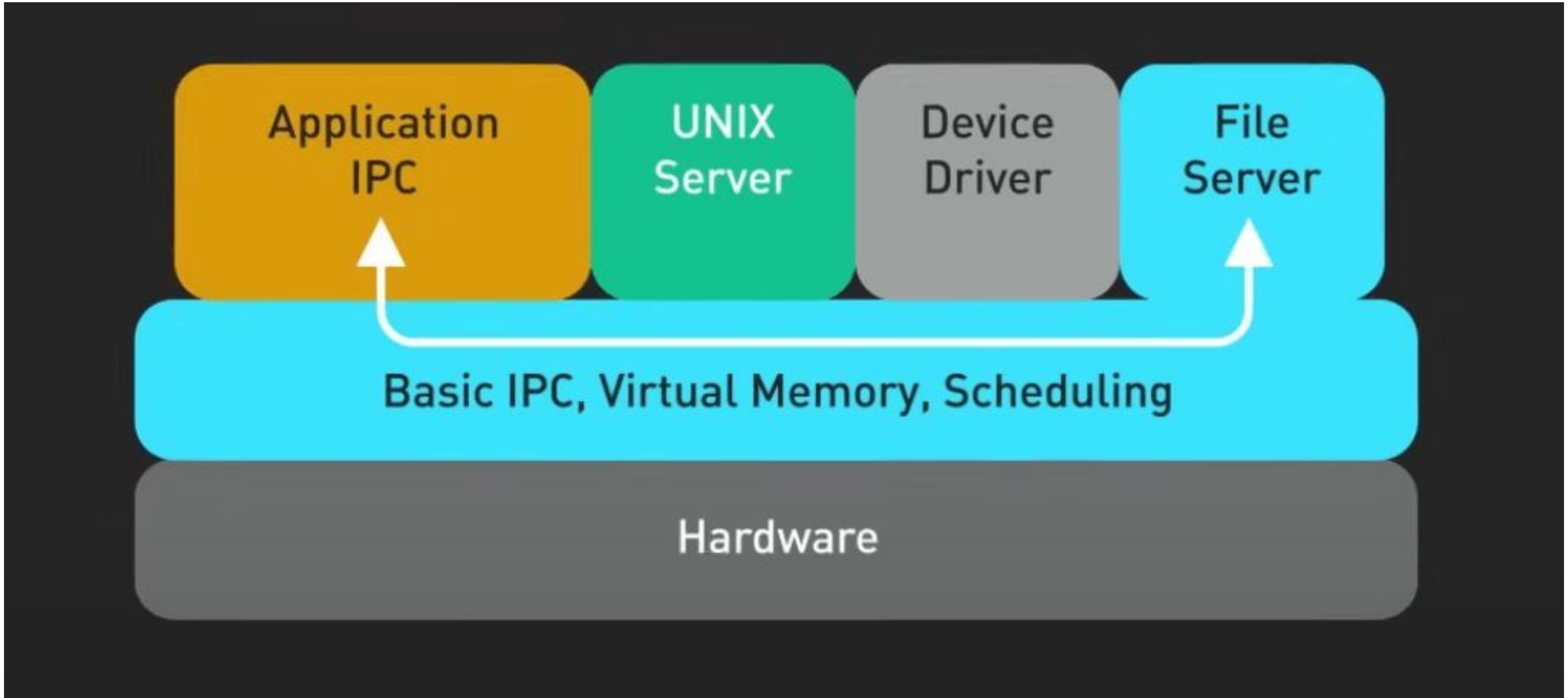


3. Microkernel Architecture

- Has a small core (kernel) with extensible plugins.
- Pros: Easy to extend and customize.
- Cons: Requires well-designed core, can become complex with many plugins.
- Use cases: IDEs (Visual Studio Code, Eclipse), operating systems.

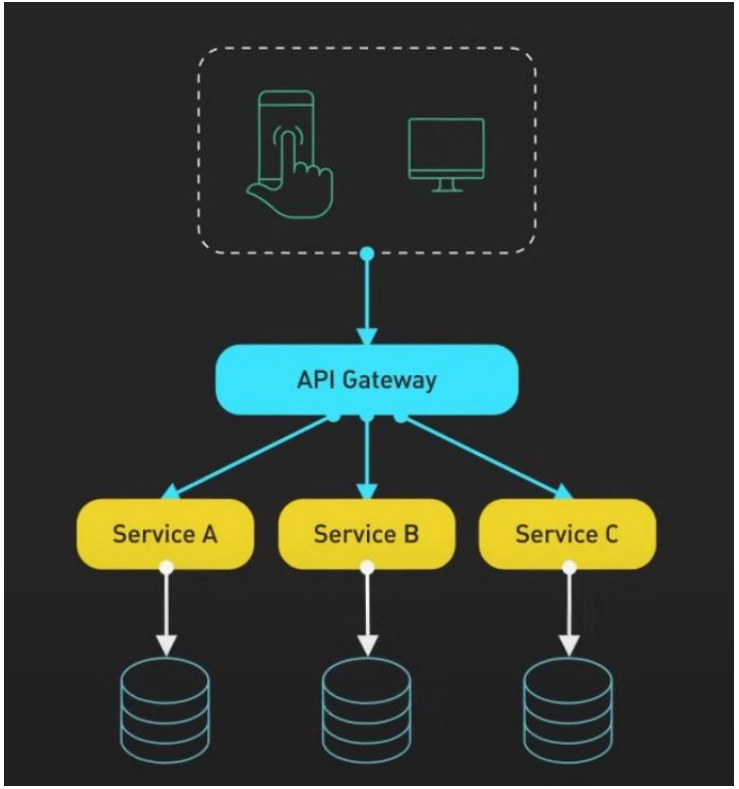


3. Microkernel Architecture



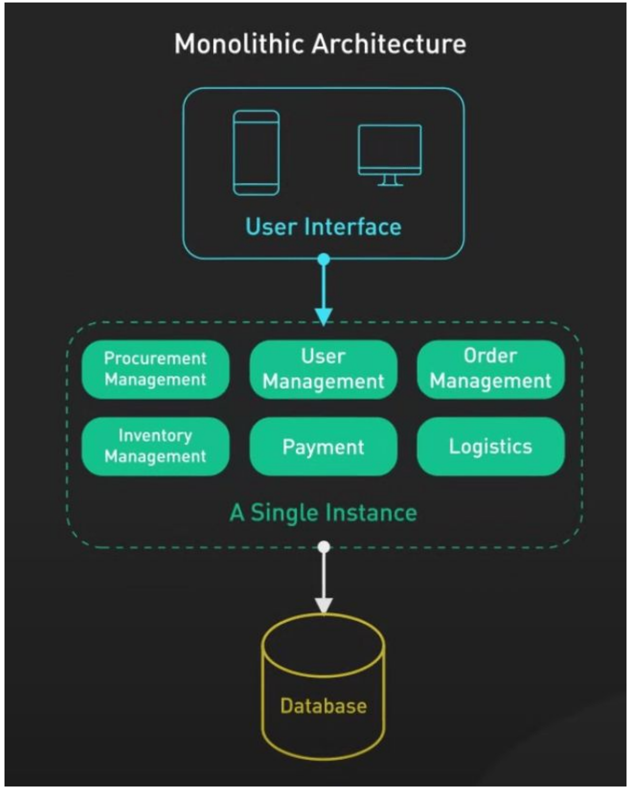
4. Microservices Architecture

- The application is divided into small, independent services communicating via APIs.
- Pros: Scalable, independent deployment, flexible technology choices.
- Cons: Complex management, requires distributed systems and monitoring.
- Use cases: Large-scale systems, e-commerce platforms, SaaS applications.



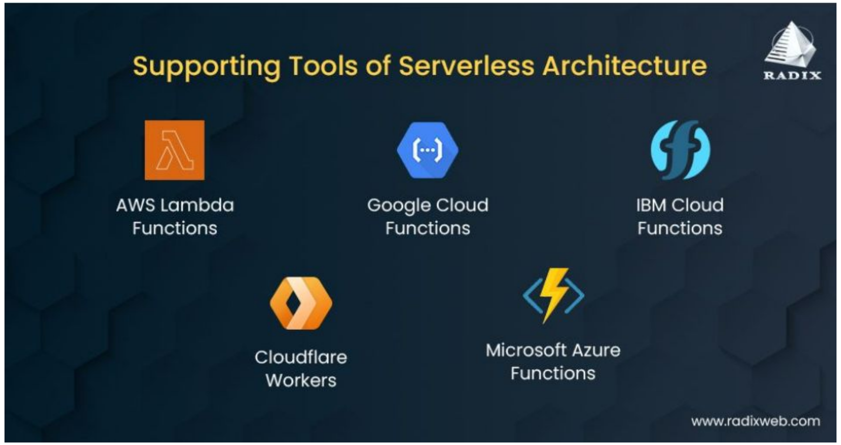
5. Monolithic Architecture

- The entire application is built as a single unit, containing both frontend and backend.
- Pros: Easy to develop, deploy, and debug.
- Cons: Difficult to scale, maintain, and implement CI/CD.
- Use cases: Small systems, startups, internal applications.



6. Serverless Architecture

- Runs on cloud platforms without managing servers, using small functions (FaaS - Function as a Service).
- Pros: Auto-scaling, cost-efficient, no infrastructure management.
- Cons: Vendor lock-in, difficult performance control.
- Use cases: Event-driven processing, lightweight applications.



Thank you