Architecture	Use When	Avoid When
Monolithic	- Building a simple or small application The team is small and the application has low complexity.	- Application is growing complex and hard to maintain You need independent deployment/scalability.
Microservices	Need to scale horizontally. Teams need to work independently. Different parts require different technology stacks.	- Managing multiple services adds too much complexity The application is simple or small.
Event-Driven	- Real-time processing is required You need to decouple services Handling high volume of data/events.	- System complexity increases with more event handlers Low-latency is not required.
Serverless	- Handling variable or unpredictable traffic Quick deployments and reduced infrastructure management Pay-as-you-go pricing model.	- High-performance, long-running tasks are needed. - Vendor lock-in and cold start issues may occur.
Hexagonal	- Need to keep business logic decoupled from external systems You want to facilitate testing and adaptability.	- The added complexity is unnecessary for small or simple applications When simplicity is paramount.