Enterprise Architecture Guide

Enterprise Architecture Guide for Cloud-Native Systems

ABSTRACT:
This document provides comprehensive guidance for building scalable, resilient cloud-native applications using modern architecture patterns.

CHAPTER 1: MICROSERVICES ARCHITECTURE

Microservices represent a paradigm shift from monolithic applications.
Key principles include:

- Single responsibility per service
- Database per service pattern
- Decentralized governance
- Failure isolation
- Technology diversity

Design Patterns:

- API Gateway for request routing
- Service Discovery with Consul/Eureka
- Circuit Breaker with Hystrix
- Bulkhead pattern for resource isolation
- Saga pattern for distributed transactions

CHAPTER 2: CONTAINER ORCHESTRATION

Kubernetes has become the de facto standard for container orchestration.
Essential concepts:

- Pods as the smallest deployable units
- Services for network abstraction
- Ingress controllers for external access
- ConfigMaps and Secrets for configuration
- Persistent Volumes for stateful workloads

Advanced Topics:

- Custom Resource Definitions (CRDs)
- Operators for application lifecycle
- Service mesh with Istio
- GitOps with ArgoCD/Flux

CHAPTER 3: DATA MANAGEMENT

Modern applications require sophisticated data strategies:

- CQRS for read/write separation
- Event sourcing for audit trails
- Polyglot persistence for optimal storage
- Distributed caching with Redis
- Database replication and sharding

Performance Optimization:

- Connection pooling (PgBouncer/HikariCP)
- Query optimization techniques
- Index strategies for different workloads
- Cache invalidation patterns
- Read replicas for scaling reads

CHAPTER 4: SECURITY BEST PRACTICES

Zero-trust security model implementation:

- Identity and Access Management (IAM)
- OAuth2/OIDC for authentication
- mTLS for service-to-service communication
- Network policies and segmentation
- Secrets management with Vault

Container Security:

- Image vulnerability scanning
- Runtime security monitoring
- Pod security policies
- RBAC implementation
- Security contexts and capabilities

CHAPTER 5: OBSERVABILITY

Three pillars: Metrics, Logs, and Traces

- Prometheus for metrics collection
- Grafana for visualization
- Jaeger/Zipkin for distributed tracing
- Fluentd/Fluent Bit for log shipping
- Alertmanager for notification routing

SLI/SLO Definition:

- Service Level Indicators (SLIs)
- Service Level Objectives (SLOs)
- Error budgets and burn rates
- Incident response procedures
- Post-mortem processes