Backend Systems Book Structure

Legend:

Green = completed

Red = important notes

**Phase 1 (we do this first)**

For the book I will use JS as the primary language. AWS as the primary cloud technology.

Source Code Management and Code Quality

Git workflows: Git Flow, trunk-based development

Introduction to Information System Architecture

**JIRI:** Monolith, layered architecture, microservices, event-driven systems, “command-driven systems”

**JIRI:** Current trends: cloud-native, serverless

Role of AI agents (e.g., request routing, recommendations, observability). We merely touch this technology with Open AI and/or MS Semantic Kernel. I will prepare something as a demo.

Fundamentals of Backend Architecture Design

Service layers: API, business logic, data access

Architectural principles: separation of concerns, DRY, SOLID

Common services: validation, authentication, authorization, configuration, error handling, logging. Examples with Express JS and Python Django.

Authentication and Authorization

Authentication strategies: JWT, OAuth2, SSO

Identity providers: Auth0, Azure AD, Keycloak

Authorization models: Role-based access control (RBAC), attribute-based access control (ABAC)

Security middleware and token management

Approach: Something You Know, Have, or Are. Explain physical keys.

Security in Backend Systems

Assets and their value, threat models.

Common threats and mitigation (OWASP Top 10)

Secure coding practices

Data encryption, TLS/SSL, secrets management

Security standards and compliance: SSLDC, ISO 27001, SOC 2, EU GDPR.

Testing Backend Applications

Unit, integration, and end-to-end testing

Testing frameworks and tools

Mocking, test coverage, and CI integration

Penetration testing (basic understanding) and threat modeling.

Cloud-Native Development

Statelessness, autoscaling, fault tolerance

Use of managed services (e.g., databases, queues, caches). These things described from cloud development angle.

Observability: logging, metrics, tracing

AWS SDK

Containerization and Virtualization

Docker basics: images, containers, volumes, networks

Orchestration overview: Docker Compose vs. Kubernetes. For example, two services in two containers orchestrated by Kubernetes.

CI/CD pipelines with containers

Deployment and Delivery

Deployment strategies: rolling, blue-green, canary

CI/CD tools: GitHub Actions, GitLab CI, Jenkins

Infrastructure as Code: Terraform, Pulumi

Code Quality

Static code analysis, linters, and code review

Refactoring and handling technical debt. Reverse-engineering.

Maintenance and Lifecycle Management

API versioning and backward compatibility

JIRI: Database migrations and schema evolution

Documentation standards (OpenAPI/Swagger, ADRs)

Long-term maintainability and scalability

Application Retirement

**Phase 2 (we need additional expertise to cover the following topics)**

edge computing (embedded software, IoT – Hardwario)

DevOps and Operational Practices

Monitoring and alerting (Prometheus, Grafana, ELK)

Tracing and log aggregation

SLOs, SLIs, SLAs

Incident management and postmortems