Project Kushim: System Architecture Design (SAD)

Lead Solutions Architect

January 2026

Contents

1 Introduction

The System Architecture Design (SAD) document defines the structural patterns, technology stack, and data flow of the Kushim platform. This design ensures the system is resilient, scalable, and maintainable.

2 Architectural Goals

- **Modularity:** Services are decoupled to ensure changes in one do not impact others.

- **High Availability:** Active-active deployment across multiple zones.

- **Data Integrity:** Strict schema enforcement for normalized data.

3 System Components

3.1 Client Tier (Frontend)

The client is a Next.js application utilizing React Server Components. It maintains a persistent WebSocket connection to the backend for real-time updates.

3.2 Application Tier (Backend)

The backend is structured into domain-driven modules:

1. **Identity Service:** Manages JWT issuance and RBAC.

2. **Ingestion Service:** Orchestrates "Adapters" that fetch data from external sources.

3. **Notification Service:** Manages WebSocket rooms and message broadcasting.

3.3 Persistence Tier (Data)

- **Primary DB:** PostgreSQL (Primary/Replica setup).

- **Cache/Message Broker:** Redis for inter-service communication and session storage.

4 Data Flow Patterns

4.1 Data Ingestion Flow

1. Trigger (Cron or Webhook) hits the Ingestion Service.

2. Specific `SourceAdapter` fetches data.

3. Data is validated and transformed into Kushim-Standard JSON.

4. Record is saved to PostgreSQL and an event is emitted to Redis.

5 Deployment Strategy

The system is containerized using Docker. Deployment is handled via a CI/CD pipeline targeting a Kubernetes cluster, utilizing Helm charts for configuration management.