Project Kushim: Data Modeling & Schema Documentation

Data Architecture Team

January 2026

Contents

1 Introduction **2**

2 Architectural Goals **2**

3 System Components **2**
3.1 Client Tier (Frontend) 2
3.2 Application Tier (Backend) 2
3.3 Persistence Tier (Data) 2

4 Data Flow Patterns **2**
4.1 Data Ingestion Flow 2

5 Deployment Strategy **3**

1 Data Architecture Overview

Kushim utilizes PostgreSQL 15+ to manage structured and semi-structured data. The design prioritizes 3rd Normal Form (3NF) for core entities while utilizing GIN-indexed JSONB for aggregated data.

2 Data Dictionary

2.1 Table: users

Column	Type	Description
id	UUID (PK)	Primary unique identifier (Auto-gen).
email	VARCHAR(255)	Unique user email for login.
password_hash	TEXT	Argon2 hashed password.
role_id	UUID (FK)	Reference to the roles table.
last_login	TIMESTAMP	Tracking for security audits.

2.2 Table: unified_records

Column	Type	Description
id	UUID (PK)	Unique record identifier.
user_id	UUID (FK)	Ownership reference for multi-tenancy.
source_id	UUID (FK)	Origin source (e.g., GitHub, Jira).
payload	JSONB	The actual data object. Indexed via GIN.
checksum	VARCHAR(64)	SHA-256 hash to prevent duplicate ingestion.

3 Indexing Strategy

To maintain sub-second performance, the following indexes are mandatory:

- `CREATE INDEX idx_records_user_source ON unified_records(user_id, source_id);`
- `CREATE INDEX idx_records_payload_search ON unified_records USING GIN (payload);`
- `CREATE INDEX idx_audit_created ON activity_logs(created_at DESC);`

4 SQL Initialization Script

```sql
-- Core Extension
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

-- Table Definition Example
CREATE TABLE data_sources (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID NOT NULL,
    provider_name VARCHAR(50) NOT NULL,
    credentials_encrypted JSONB,
```

```
    status VARCHAR(20) DEFAULT 'active',
    created_at TIMESTAMPTZ DEFAULT NOW()
);
```