Sutradhara

Comprehensive Design and Requirements Document

1. Introduction

Sutradhara is a generic, domain-agnostic AI data access platform designed to bridge natural language user queries and structured relational databases. The platform enables users to retrieve meaningful insights from enterprise data using plain language while enforcing strict governance, access control, and security boundaries. Sutradhara is designed for multi-domain usage including education, insurance, ecommerce, inventory, HR, and finance systems.

2. Problem Statement

Modern organizations store large volumes of data in relational databases that are difficult to access for non-technical users. Existing solutions either expose raw data insecurely or tightly couple AI logic to database schemas. Sutradhara solves this by introducing a policy-driven, schema-aware, and LLM-assisted orchestration layer that ensures safe and explainable data access.

3. Vision and Objectives

The vision of Sutradhara is to act as an intelligent orchestrator that understands user intent, enforces organizational rules, and delivers precise, summarized answers. Objectives include domain independence, zero-trust AI interaction, configuration-driven onboarding, and enterprise-grade observability.

4. Core Design Principles

Key principles include zero direct database access by LLMs, strict separation of concerns, metadata-driven domain configuration, least-privilege data exposure, and full auditability of decisions and data flows.

5. High-Level Architecture

The architecture consists of multiple decoupled layers: API Gateway, Identity and Context Resolver, Intent Resolution Agent, Semantic Mapping Engine, Policy and Authorization Engine, Data Retrieval Engine, and LLM Summarization Layer. Each layer performs a single responsibility and communicates via well-defined contracts.

6. Identity and Context Management

User identity is established using JWT tokens that contain user identifiers, roles, organization context, and linked entities. This context persists throughout request processing and is used to enforce row-level and column-level security policies.

7. Intent Resolution Layer

This layer uses an LLM to transform natural language questions into structured intent objects. Intents include requested action, logical entity, filters, and temporal constraints. The output format is strictly validated JSON to avoid ambiguity.

8. Schema and Semantic Configuration

Domains are onboarded by registering database connections, schemas, tables, and column metadata. Logical entities are mapped to physical tables through configuration files that define semantic meaning, ownership relationships, and join paths.

9. Policy and Access Control Engine

The policy engine evaluates whether a resolved intent is allowed for the requesting user. Policies define role-based access, entity-level permissions, column visibility, and row-level filtering rules. Unauthorized access attempts are blocked before data retrieval.

10. Data Retrieval Engine

This engine generates safe, parameterized SQL queries based on validated intent and policy outputs. It enforces limits on result size, masks sensitive columns, and ensures that only authorized data is retrieved.

11. LLM Summarization Layer

The summarization layer receives only sanitized datasets along with the original user question. The LLM generates human-readable summaries, explanations, or insights without exposure to internal identifiers or schemas.

12. Multi-Domain Usage Examples

Examples include parents viewing student attendance in school systems, customers reviewing insurance policies, shoppers checking order history in ecommerce platforms, and managers inspecting inventory levels.

13. Open Source Tools and Ecosystem

Sutradhara can be implemented using open-source components such as Open Data QnA for intent-to-SQL workflows, MindsDB Open Source for data connectors, LangGraph for agent orchestration, Open Policy Agent for policy enforcement, and Apache Superset metadata concepts for schema management.

14. Non-Functional Requirements

Non-functional requirements include high availability, horizontal scalability, low latency response times, strong security posture, observability through logging and metrics, and compliance readiness.

15. Recommended Implementation Workflow

A typical workflow includes registering the domain, configuring schema metadata, defining policies, deploying intent resolution agents, integrating an LLM provider, and onboarding end users.

16. Conclusion

Sutradhara provides a robust, secure, and extensible foundation for AI-driven access to enterprise data. Its generic and policy-driven architecture makes it suitable for a wide range of industries and use cases.