AI Incident Assistant - System Design Document

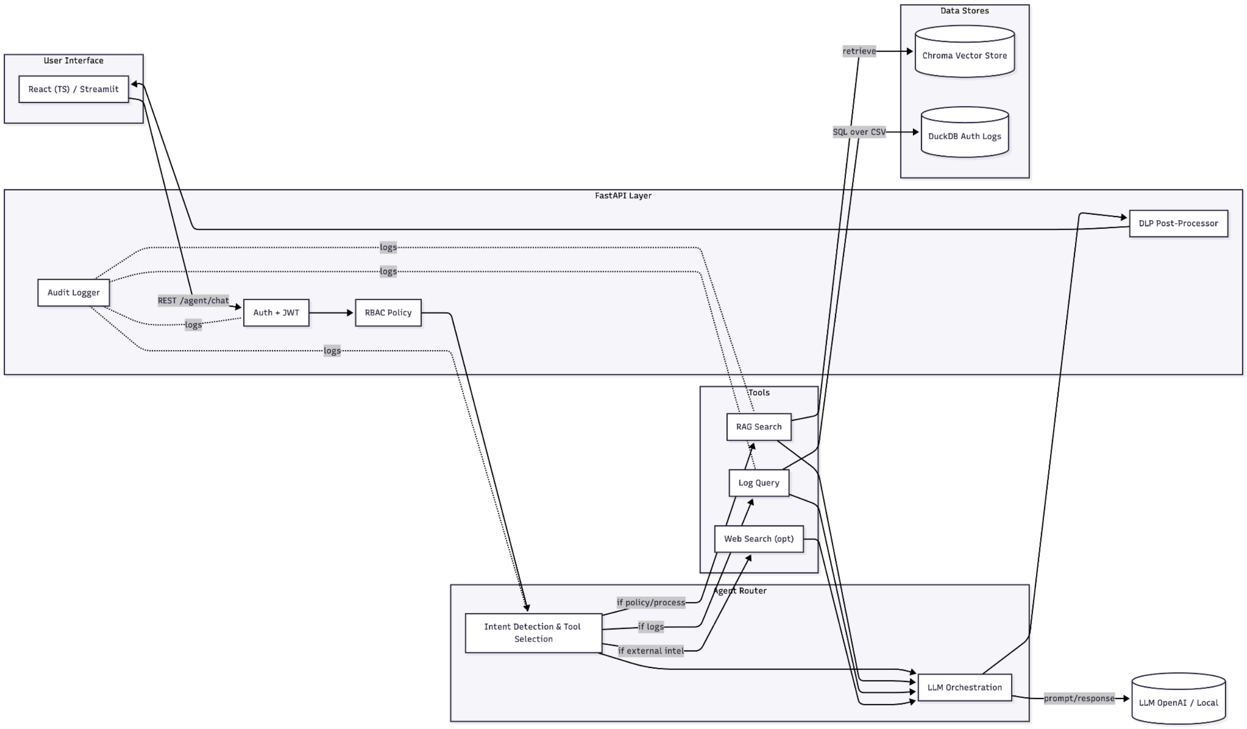
1. High-Level Architecture

Overview

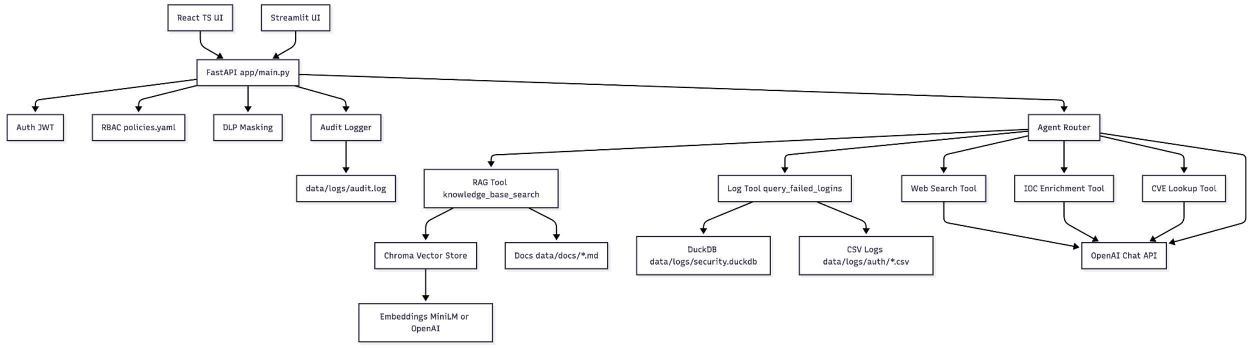
The **AI Incident Assistant** is an agentic security assistant that helps teams investigate incidents, query logs, and access internal knowledge bases securely. It integrates three main capabilities:

- **Retrieval-Augmented Generation (RAG)** for contextual Q&A from internal documentation.
- **Structured Tools** for querying logs, enriching Indicators of Compromise (IOCs), and looking up CVEs.
- **Agent Layer** that decides which tool(s) to invoke for each query, combining results with LLM reasoning.

Architecture Diagram



Main Components



2. Code Organization

Directory	Purpose
`/app`	FastAPI backend with routes, LLM agent, and tools
`/app/tools`	Independent tool modules (`log_query`, `ioc`, `cve`, `web_search`)
`/frontend`	Streamlit-based prototype frontend
`/frontend-react-ts`	React + TypeScript frontend (production UI)
`/data/docs`	Knowledge base markdown files for RAG
`/data/logs`	Log files and DuckDB database
`/scripts`	Bootstrap and ingest scripts to initialize data and Chroma index
`/docker`	Dockerfile and Compose setup for containerized deployment
`README.md`	Setup and usage instructions

3. Modes of Operation

1. GitHub Codespaces Mode (Development / Demo)

- Lightweight, CPU embeddings (MiniLM-L6-v2)

- Run FastAPI and Streamlit apps directly
- OpenAI LLM optional (uses fallback rules if API unavailable)

2. Dockerized Mode (Local / Production)

- Containers for `api`, `ui`, and optionally `web`
- Persistent Chroma & DuckDB volumes
- Pre-seeded data through bootstrap scripts
- Suitable for demonstrations, team deployments, or isolated environments

3. Headless API Mode

- Run API only for programmatic integration (CLI, monitoring bots, etc.)

---

4. Main Use Cases Implemented

Use Case	Example Query	Tool(s) Used	Description
Log Analysis	“Show me today’s failed login attempts for username jdoe”	Log Query	Retrieves and summarizes failed logins using DuckDB.
Policy / Playbook Lookup	“How should I handle a phishing email?”	RAG	Retrieves relevant playbook markdowns and summarizes steps.
CVE Intelligence	“What are critical TLS vulnerabilities this month?”	CVE Lookup + RAG	Searches CVE data and synthesizes summaries.
IOC Enrichment	“Investigate IP 185.21.54.100”	IOC Tool	Returns IP reputation, ASN, and TOR/blacklist info.
Multi-tool Query (Agentic)	“Check today’s failed logins and list recent CVEs related to TLS”	Log Query + RAG + CVE	Combines multiple tools automatically.

---

5. Areas for Improvement and Clarification

To further mature the prototype for a final assessment, focus on demonstrating the full implementation of the security features and completing the stretch goals.

Focus Area	Suggested Improvement	Rationale
Prompt Injection Defense	Use a multi-stage defense: a fast keyword filter **plus** a small LLM-based classifier to re-phrase or block more sophisticated attacks.	The current heuristic filter is necessary but can be bypassed by a determined attacker.
Audit Logging	Implement a **structured JSON** format capturing every tool call, inputs/outputs, the agent decision, and final answer; store in a dedicated `audit_logs/` volume.	Essential for traceability in a security product; required to meet the audit requirement fully.
Security Transparency (Stretch)	Modify API responses to include a `transparency_info` field listing which tools ran and which documents were consulted.	Increases user trust and explains “why did I answer this way?”
Conversation Memory (Stretch)	Integrate **Buffer Memory** (e.g., LangChain) for multi-turn context (“What about user smith?”).	Improves conversational flow and usability for investigations.
Test Coverage	Add unit/integration tests for DLP, injection defense, and RBAC logic; include golden tests for retrieval and agent routing.	Critical in a security context to ensure controls are fail-safe.

Additional enhancements: hybrid retrieval (BM25 + dense), semantic re-ranking, retry/backoff for LLM calls, dynamic RBAC from IAM, and a unified React UI with tool tables and citations.