Reginald – System Architecture & Deployment Whitepaper

Introduction

Reginald is an AI Copilot designed to support financial crime and compliance analysts. By providing instant, AI-driven responses to regulatory and investigative questions, Reginald streamlines compliance operations and reduces the burden of manual research. Its architecture combines a modern frontend interface, a powerful backend, and seamless integration with external AI services to deliver reliable and secure functionality. The following paper outlines the system's design and deployment, describing how its components work together to create a comprehensive platform for compliance professionals.

System Overview

At its core, Reginald connects users to an AI-powered chat interface that simulates the feel of a natural conversation. When analysts log in, they are greeted by a responsive dashboard and can immediately begin interacting with the chatbot. Every query is routed through the backend, authenticated, and then securely passed to the Gemini API for AI inference. Responses are streamed back to the user in real time using Server-Sent Events (SSE), allowing the system to display answers word by word rather than waiting for the entire message to generate. This design ensures that the system feels fast, conversational, and highly responsive. The architecture consists of the following major components: the frontend built with React and ShadCN UI, the FastAPI backend, a shared module library, a fully implemented database for persistence, and external AI services provided by Gemini. Together, these pieces form a cohesive and scalable system that balances user experience with technical robustness.

Frontend

The frontend is implemented using React with Vite as the build tool, styled with Tailwind v4 and the ShadCN UI library. This combination ensures both rapid development and a clean, modern user interface. The primary user flows include secure login, dashboard navigation, and chat interaction. The frontend also handles streaming updates from the backend, progressively displaying AI responses in a way that mirrors natural conversation. By focusing on simplicity and responsiveness, the interface empowers compliance analysts to focus on their work without being distracted by technical complexity.

Backend

The backend is built with FastAPI, a modern Python web framework optimized for speed and developer productivity. It is responsible for authentication, routing, and the orchestration of AI interactions. When

a user sends a query, the backend validates the session, logs the interaction, and relays the request to the Gemini API. Responses are then streamed back through the backend, which ensures that all communication remains secure and consistent. Because it runs on Heroku, the backend benefits from managed deployment, automatic scaling, and integrated monitoring without the overhead of container orchestration platforms.

Database Layer

Unlike earlier iterations of the system, the database layer is fully implemented in this version of Reginald. The database stores user information, chat histories, audit logs, and case-tracking records. This persistence is vital for compliance operations, as it enables analysts to revisit past conversations, demonstrate regulatory adherence, and maintain transparent audit trails. The database ensures that Reginald not only provides real-time assistance but also serves as a lasting resource for compliance teams.

Deployment

Reginald is hosted on Heroku, which provides a reliable and secure environment for both the backend and frontend. Heroku's managed platform allows the application to scale with demand while simplifying operational concerns such as SSL, logging, and monitoring. The frontend is built and deployed as part of the same pipeline, ensuring a seamless integration between the user interface and the backend services. Secrets, API keys, and configuration values are stored securely using Heroku's environment management system, which aligns with compliance best practices. This hosting model ensures that Reginald remains accessible, resilient, and easy to maintain.

Security and Reliability

Security and reliability are foundational to Reginald's design. All traffic is encrypted via HTTPS, and authentication is managed with JWT tokens to ensure only authorized users can access the system. The backend validates all inputs and enforces strict ownership rules on chats and documents. Streaming responses are handled in a way that minimizes latency and improves the perceived responsiveness of the system. Furthermore, the database adds fault tolerance by persisting critical data, ensuring that user sessions and records are never lost.

Conclusion

Reginald represents a modern approach to compliance assistance, combining AI-driven insights with a secure and resilient system architecture. By uniting a responsive frontend, a scalable backend, a persistent database, and managed deployment on Heroku, the system offers compliance analysts both immediacy and reliability. As regulations evolve and compliance workloads grow more complex,

Reginald provides a robust foundation that can adapt and scale, ensuring that compliance teams remain supported and effective in their work.