# AI-Powered WhatsApp Automation Platform System Architecture Overview

This document describes the system architecture for an AI-powered WhatsApp automation platform. The solution is designed to help businesses automate customer support and inquiries using artificial intelligence combined with a structured knowledge base. The architecture follows modern SaaS best practices to ensure scalability, security, and maintainability.

## High-Level Architecture

The platform is composed of four major layers: a web-based admin dashboard, a secure backend API, an AI and automation engine, and a WhatsApp integration layer. Each layer is independently scalable and communicates through well-defined interfaces.

## Core Design Principles

The system uses a WhatsApp account–centric architecture. Each WhatsApp number represents a single business bot with its own knowledge base, AI configuration, analytics, and automation rules. Multiple users can collaborate on the same WhatsApp account using role-based permissions.

## Authentication and Authorization

User authentication is handled via Supabase Auth, providing secure email and password login with JWT-based sessions. Authorization is enforced in the backend using role-based access per WhatsApp account. Roles include Owner, Admin, Operator, and Viewer, each with clearly defined permissions.

## Knowledge Base and AI (RAG Pipeline)

Administrators upload business documents such as PDFs or text files through the dashboard. These documents are parsed, chunked, and embedded using an AI embedding model. When a WhatsApp message is received, the system performs semantic search on the knowledge base to retrieve relevant context, which is then used by the AI model to generate accurate and grounded responses.

## WhatsApp Integration

The platform integrates with WhatsApp using a persistent web session. QR-based authentication is used during onboarding, and session handling logic ensures reliability and automatic reconnection. Each WhatsApp account maintains a single active session to avoid conflicts.

## Admin Dashboard

The admin dashboard is built with Next.js and TypeScript and provides functionality for WhatsApp connection management, knowledge base administration, conversation monitoring, analytics, AI configuration, and team access management. The user interface supports dark mode and responsive layouts.

## Scalability and Future Expansion

The architecture is designed to support future expansion, including multi-tenant SaaS billing, multiple WhatsApp accounts per user, advanced automation workflows, CRM integrations, and plugin-based extensions. The modular design allows the platform to scale from an MVP to enterprise-level usage.

## Conclusion

This architecture provides a robust and flexible foundation for delivering a secure, scalable, and intelligent WhatsApp automation solution. It ensures clear separation of concerns, reliable AI behavior, and long-term maintainability.