

[> Menu](#)

Overview

Relevant source files

The AgenteInteligente system implements an AI-powered customer service agent specialized in Mexican tax and accounting information. The system provides an embeddable chat interface that connects to specialized AI models capable of answering questions about different tax regimes, retrieving account information, and providing personalized responses using a Retrieval-Augmented Generation (RAG) architecture.

For specific implementation details on the frontend components, see [Frontend System](#), and for details on the AI backend, see [Backend AI System](#).

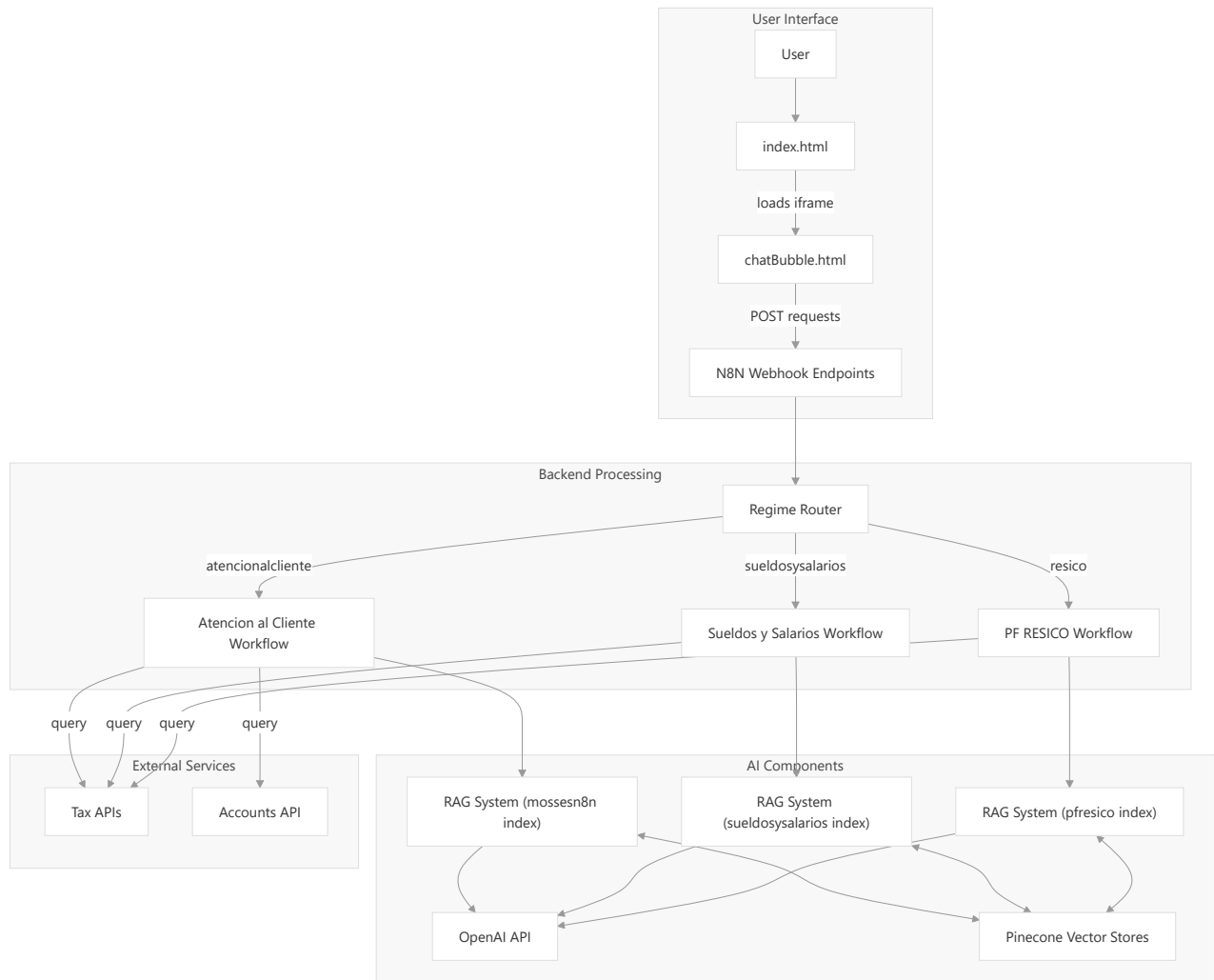
System Architecture

The system follows a client-server architecture with a JavaScript frontend chat interface that communicates with n8n workflows powering the AI backend. The architecture separates concerns between the presentation layer (chat UI), communication layer (webhooks), and processing layer (AI agents and RAG systems).

Ask Devin about axlRoman/AgenteInteligente

Deep Research





Sources: chatBubble.html | 1-573 index.html | 1-69 Atencion_al_Cliente.json | 1-1717
PF_RESICO.json | 1-1133

Core Components

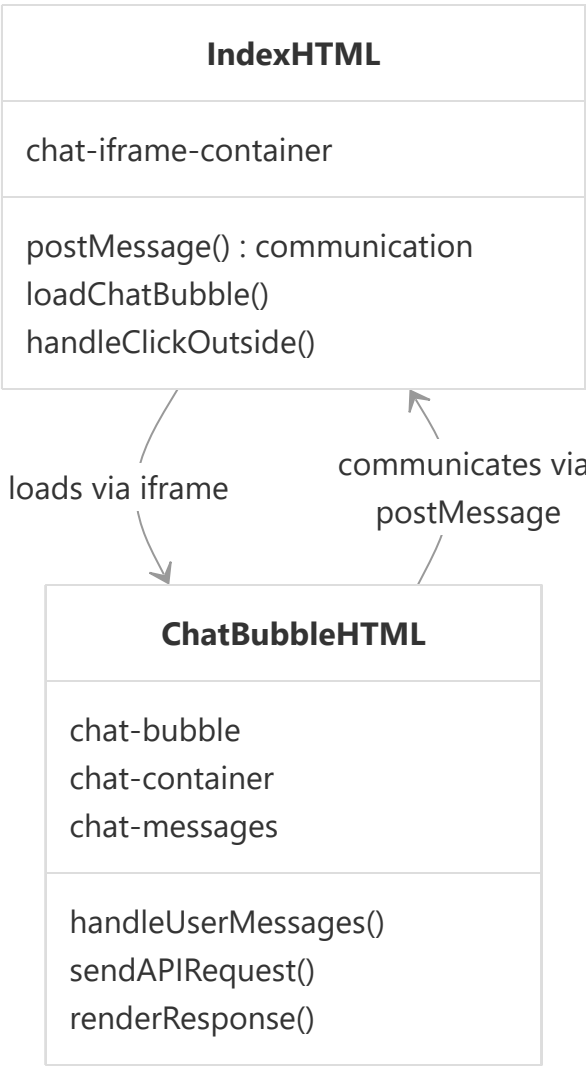
1. Chat Interface

The chat interface consists of two main HTML files:

- **Parent page (`index.html`):** Loads the chat bubble in an `iframe` and handles parent-iframe communication

Ask Devin about axlRoman/AgenteInteligente

Deep Research



Sources: index.html | 20-66 chatBubble.html | 280-570

2. Backend Workflows

The backend is implemented using n8n workflows that process user requests via webhooks. Each workflow specializes in a specific tax regime or service type:

Workflow	Purpose	Vector Store Index
Atencion al Cliente	General customer service	mossesn8n

Ask Devin about axlRoman/AgenteInteligente

Deep Research

1. Receive webhook request with user message
2. Process message using RAG architecture
3. Query relevant APIs if needed
4. Format and return response

Sources: `Atencion_al_Cliente.json` | 2-19 `PF_RESICO.json` | 2-19
`Atencion_al_Cliente.json` | 323-339

3. Retrieval-Augmented Generation System

The AI engine uses a RAG architecture combining vector search with generative AI to produce accurate, contextually relevant responses.

Ask Devin about axlRoman/AgenteInteligente

Deep Research



Ask Devin about axlRoman/AgenteInteligente

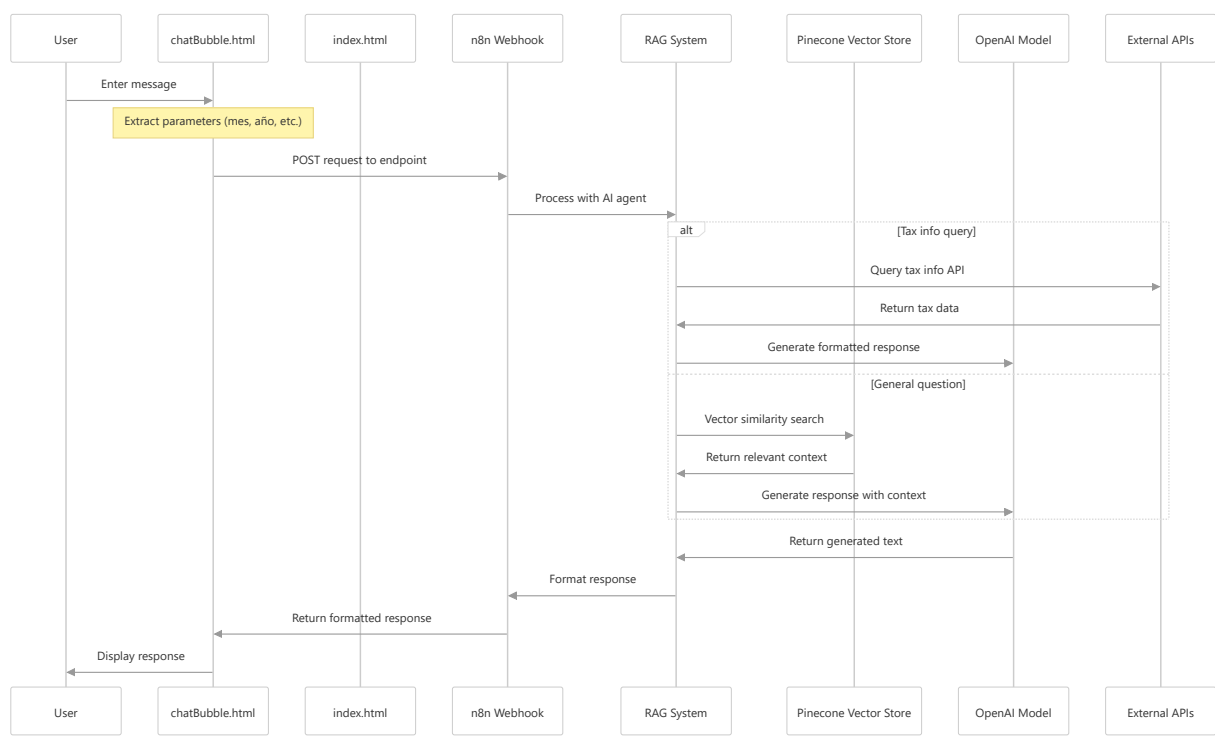
Deep Research ☐

The system uses three specialized Pinecone vector stores (mossesn8n, sueldosysalarios, pfresico) to store domain-specific knowledge.

Sources: `Atencion_al_Cliente.json` | 192-218 `PF_RESICO.json` | 120-136 `PF_RESICO.json` | 702-710

Communication Flow

The system's communication flow shows how user messages are processed through the various components:



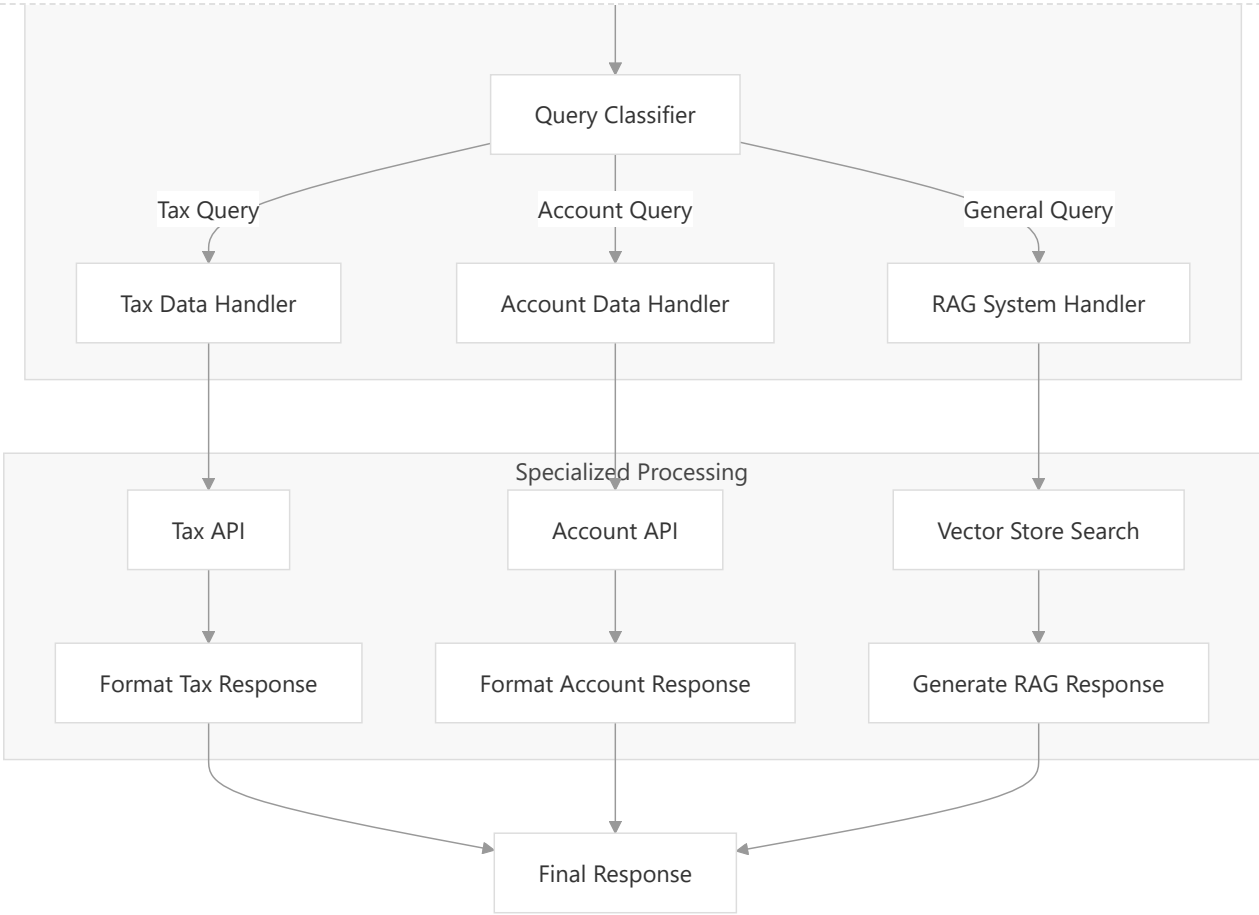
Sources: `chatBubble.html` | 460-508 `Atencion_al_Cliente.json` | 1383-1415
`Atencion_al_Cliente.json` | 547-558

Query Processing Logic

Ask Devin about axlRoman/AgenteInteligente

Deep Research

2. **Accounts receivable queries:** Questions about accounts receivable are routed to a specialized API endpoint that returns account information.
3. **General queries:** Other questions are handled through the RAG system, which retrieves relevant information from the vector store and generates contextually appropriate responses.



Sources: Atencion_al_Cliente.json | 429-459 Atencion_al_Cliente.json | 414-448

PF_RESICO.json | 453-498

Deployment Architecture

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐

2. **Backend n8n server:** Running the workflow automation platform that powers the AI agent logic
3. **External APIs:** Tax information and account data APIs

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐