

> Menu

Frontend System

Relevant source files

The Frontend System provides a web-based chat interface that allows users to interact with the AgenteInteligente virtual assistant. This system handles the user-facing components, including the chat bubble, chat window, messaging interface, and communication with the backend AI system.

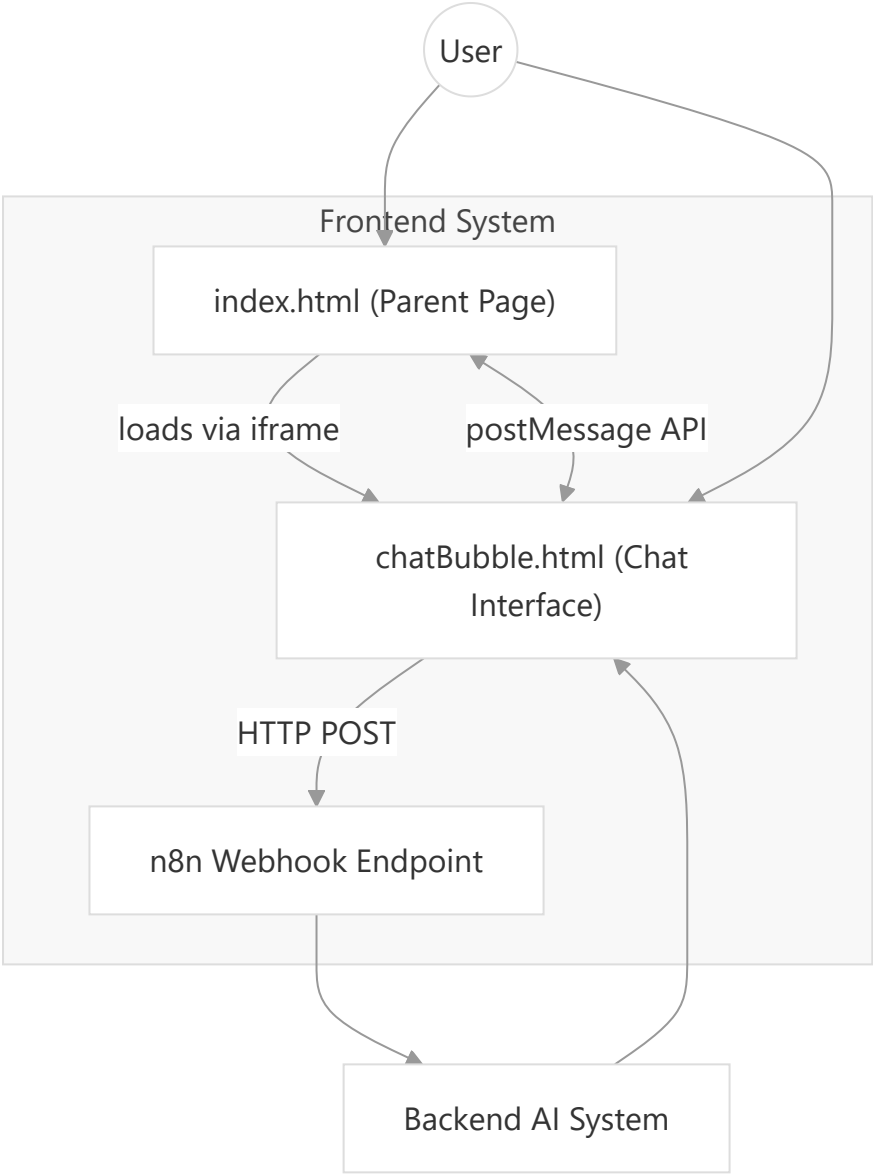
System Architecture

The frontend uses a parent-iframe architecture where a main HTML page hosts a chat interface within an iframe. These components communicate through the `postMessage` API, and the chat interface connects to the backend n8n workflows to process user queries.

Ask Devin about axlRoman/AgenteInteligente

Deep Research





Sources: `index.html` `chatBubble.html`

Core Components

The Frontend System consists of two primary components that work together to provide the chat interface:

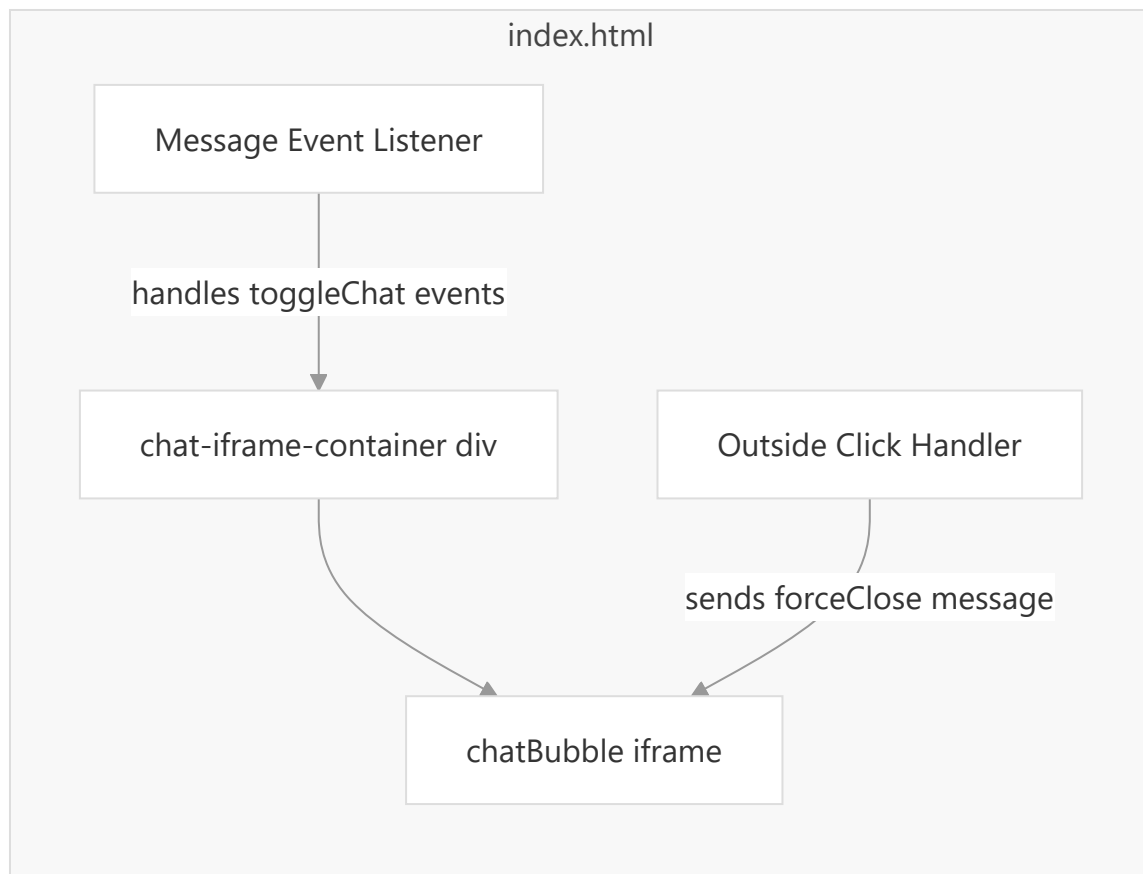
Parent Container (index.html)

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐

Key features:

- Creates a container for the chat iframe with transition effects
- Passes configuration parameters to the chat interface via URL query parameters
- Listens for messages from the iframe to adjust the container size
- Detects clicks outside the chat area to close the expanded chat window



Sources: `index.html` | 9-67

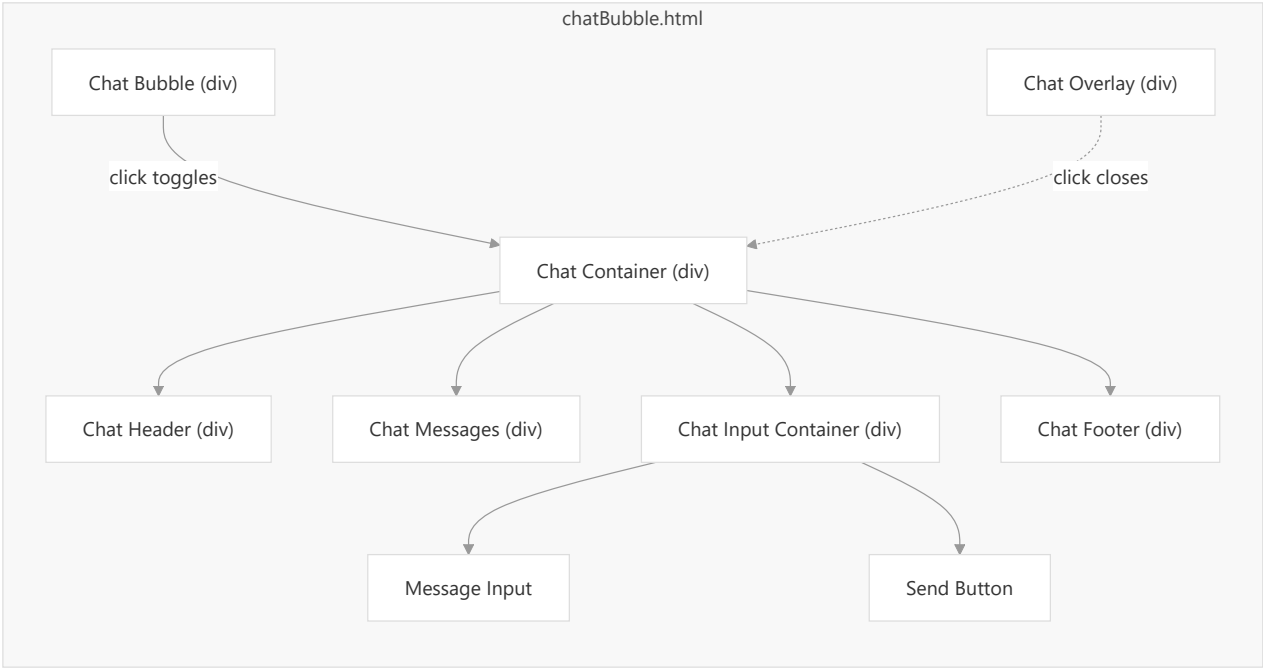
Chat Interface (`chatBubble.html`)

The chat interface provides the actual UI components that users interact with, including:

1. **Chat Bubble:** A floating button displaying the LIA logo that users click to open the chat

Ask Devin about axlRoman/AgenteInteligente

Deep Research



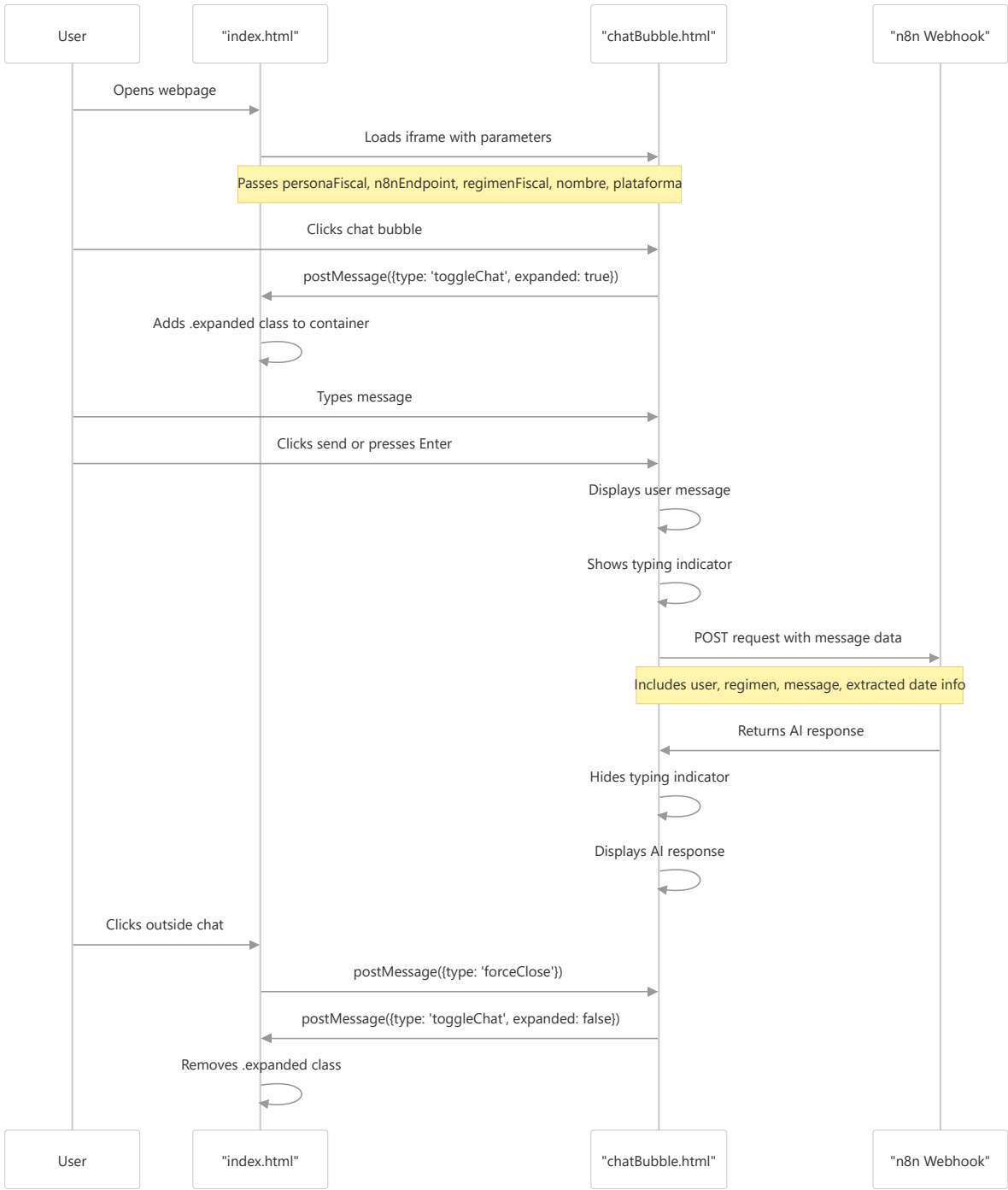
Sources: chatBubble.html | 15-189 chatBubble.html | 280-314

Communication Flow

The Frontend System components communicate through a well-defined flow:

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐



Sources: index.html | 37-66 chatBubble.html | 348-392 chatBubble.html | 460-508

Configuration Parameters

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐

Parameter	Description	Example
personaFiscal	User/taxpayer identifier	"d8f2534c-a16e-44ec-a574-7ba4f75c8f0418411"
n8nEndpoint	Backend API endpoint	" https://ai.mosses.mx/webhook/regimenes "
regimenFiscal	Tax regime code	"626"
nombre	User's name for personalization	"Axel Roman"
plataforma	Platform identifier	"me"

Sources: `index.html` | 47-54

User Interaction Handling

The chat interface implements several user interaction patterns:

1. Chat Toggling

The chat can be opened/closed through multiple methods:

- Clicking the chat bubble opens the chat
- Clicking the minimize button in the header closes the chat
- Clicking outside the chat area closes the chat
- Pressing the Escape key closes the chat

```
// Toggle chat visibility
function toggleChat() {
  const isActive = chatContainer.classList.contains('active');
  if (isActive) {
    closeChatContainer();
  } else {
    openChatContainer();
  }
}
```

Ask Devin about axlRoman/AgenteInteligente

Deep Research

```
    }, '*');  
  }  
}
```

Sources: chatBubble.html | 348-392

2. Message Sending

Messages are sent through:

- Clicking the send button
- Pressing Enter in the input field

When a message is sent:

1. The message is displayed in the chat
2. A typing indicator appears
3. The message is sent to the backend API
4. The API response is displayed when received

Sources: chatBubble.html | 460-508

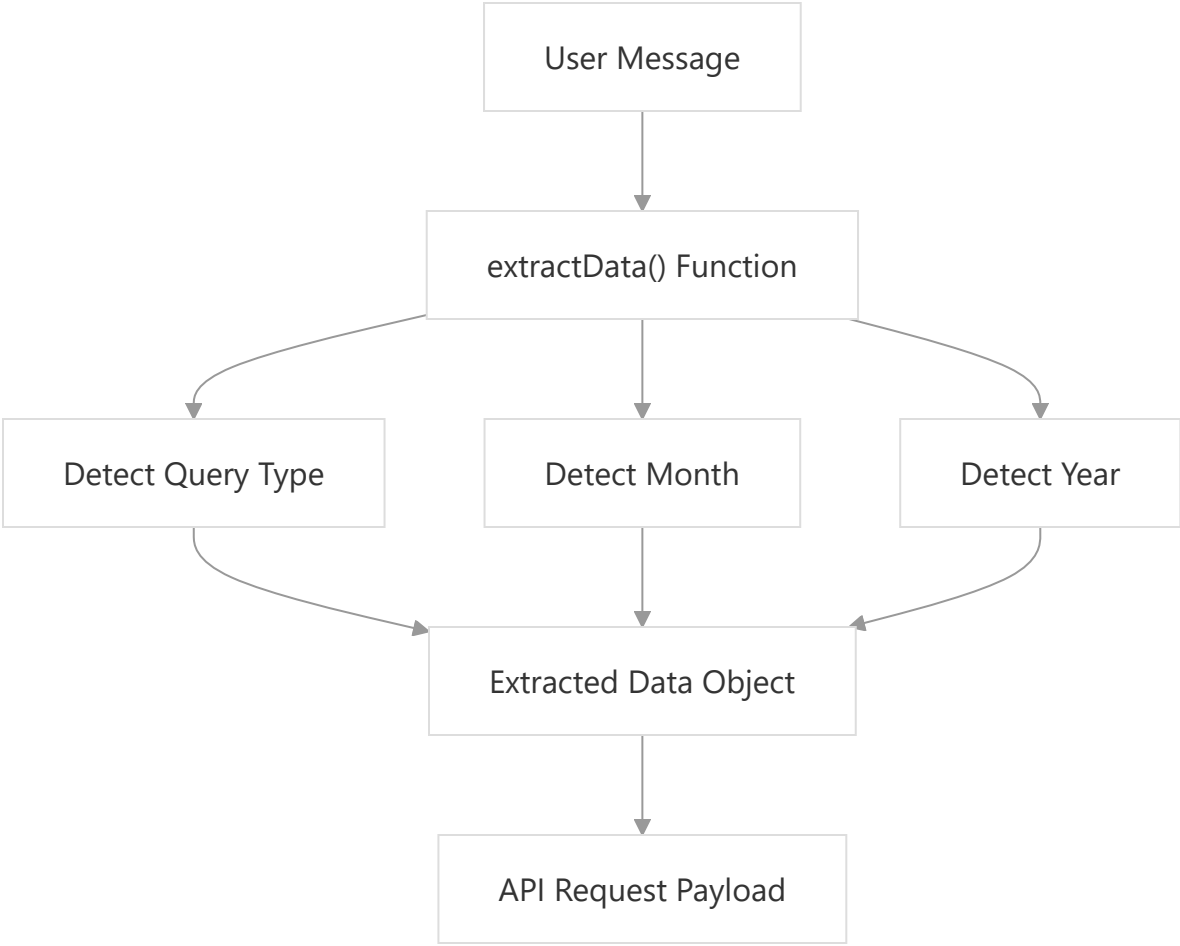
Message Processing

Data Extraction

The chat interface analyzes message content to extract relevant information:

Ask Devin about axlRoman/AgenteInteligente

Deep Research



The system extracts:

DeepWiki axlRoman/AgenteInteligente

Share



- Year references (e.g., "2023")
- End-of-month dates (for financial calculations)

Sources: chatBubble.html | 413-444

Message Formatting

The system formats messages for display:

Ask Devin about axlRoman/AgenteInteligente

Deep Research ☐

Sources: chatBubble.html | 511-541

Backend Communication

The chat interface communicates with the backend through a POST request to the n8n webhook endpoint specified in the configuration:

```
const response = await fetch(N8N_ENDPOINT, {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    user: personaFiscal,
    regimen: regimenFiscal,
    plataforma: plataforma,
    message: message,
    mes: extractedData.mes,
    nombreMes: extractedData.nombreMes,
    anio: extractedData.anio,
    ultimoDia: extractedData.ultimoDia,
    timestamp: new Date().toISOString(),
  }),
});
```

Ask Devin about axlRoman/AgenteInteligente

Deep Research