SAP Travel Assistant: Intelligent Booking with Multi-Agent Architecture

Transforming Travel Planning with GenAI

- Likith Vinayaka Giridhar

The Challenge - Our Mission

Book me a flight to Sapphire

FACILITATE SAP EMPLOYEE TRAVEL BOOKING THROUGH A CHAT INTERFACE

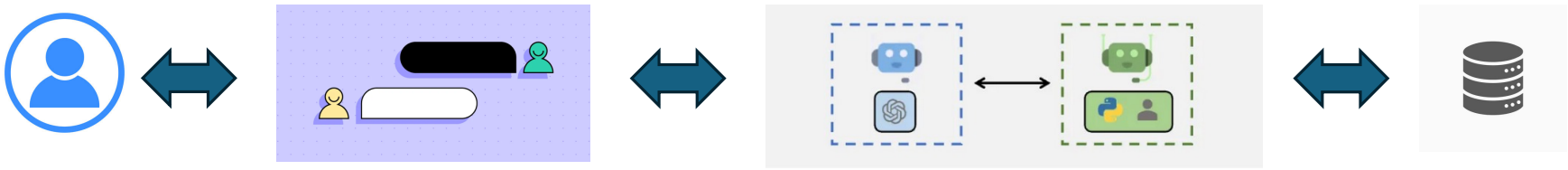
HANDLE COMPLEX QUERIES LIKE "BOOK ME A FLIGHT TO SAPPHIRE"

ENSURE COMPLIANCE WITH SAP'S TRAVEL POLICIES

PROVIDE A SEAMLESS, CONVERSATIONAL USER EXPERIENCE

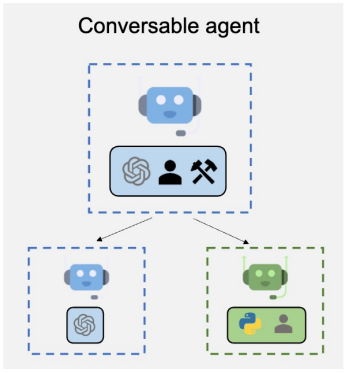
Our Solution - Overview
SAP Travel Assistant

- An AI-driven, chat-based travel booking system
- Leverages a multi-agent architecture inspired by Autogen
- Integrates SAP-specific data and policies
- Uses GPT-4-turbo for natural language interaction

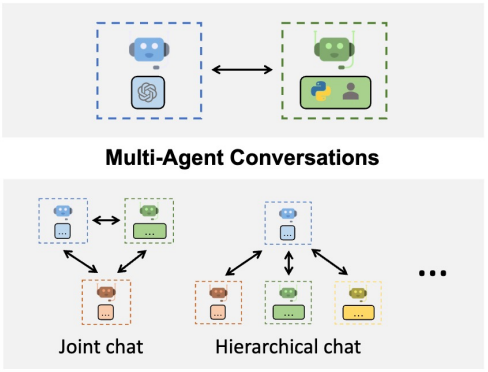


Multi-Agent Architecture

- Power of Collaboration
- Travel Agent: Books flights & accommodations
- Search Agent: Retrieves SAP internal info
- Inspired by Microsoft's Autogen framework
- Agents work together to solve complex tasks



Agent Customization



Flexible Conversation Patterns

Built a custom framework

Case for my custom framework:

Control Over Token Usage:

- By building our own framework, we have full control over the data sent to OpenAI, ensuring optimized token usage.
- We can precisely manage the prompt size, context length, and overall token consumption.

Customization:

- We custom framework is tailored to the specific requirements of the SAP travel booking use case.
- It allows for precise control over conversation flow, which is harder to achieve with a generalized framework like Autogen.

Transparency and Debugging:

- Full visibility into the codebase makes debugging and extending the solution easier.
- We can directly modify and understand every part of the interaction logic, which is beneficial for maintenance and future enhancements.

Performance:

- A custom solution can be more performant for the specific use case since it's streamlined to handle only the necessary functionality.
- Reduced overhead compared to a generalized framework like Autogen.

Case against my custom framework:

Direct Communication between agents:

- Although not as explicit as in Autogen, our agents collaborate. (Eg: The TravelAgent uses information retrieved by the SearchAgent)

State and Context preservation:

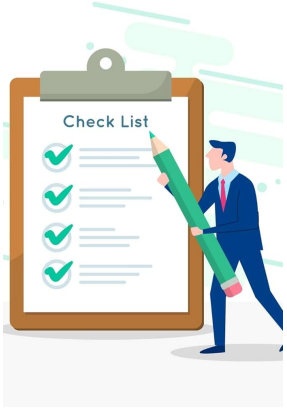
- We maintain state, particularly in our streamlit app. We track the current employee, chat history and cumulative query string ("full_query") that includes all past interactions.

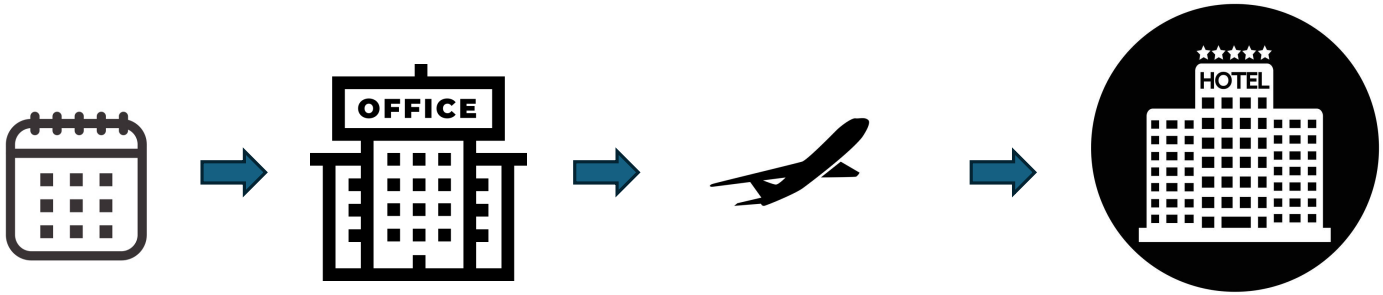
Limited Autonomy:

- Our implementation of Agents feel more like utility classes rather than autonomous agents that communicate directly.

Key Features

- Intelligent & Compliant
- Employee-specific travel options
- Departure from employee city only
- Real-time event information
- Round-trip booking capability
- Hotel suggestions near venues





Data Integration

Rich, structured datasets:

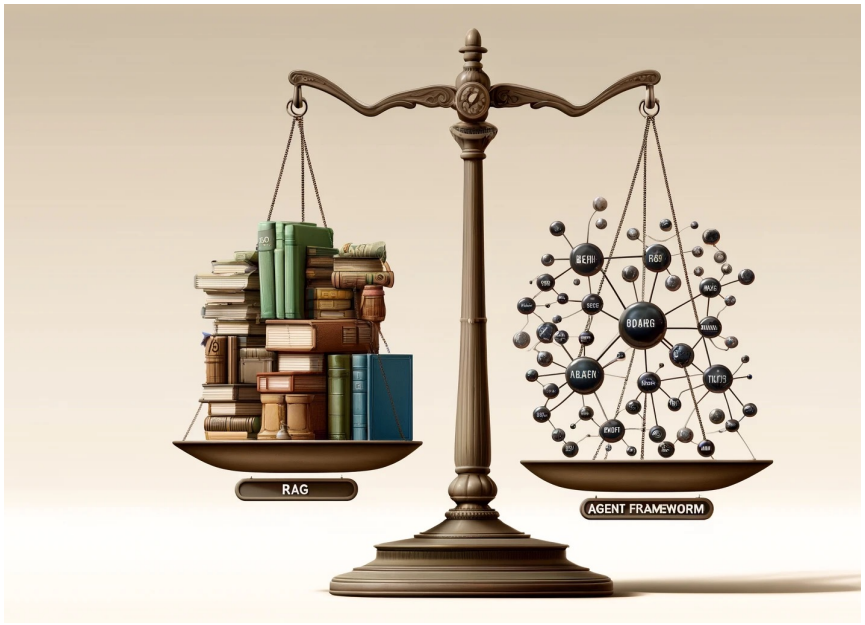
- 5 SAP employees
- 3 SAP events (Sapphire, TechEd, SuccessConnect)
- 50+ flight options
- Multiple hotel choices per event
- Ensures accuracy and relevance

OpenAI

User Experience - Conversational AI at Its Best

- Natural language queries
- Context-aware responses
- Multi-turn conversations
- Clear, actionable information
- Powered by GPT-4-turbo

Technical Innovation - Beyond RAG



- Surpasses Retrieval-Augmented Generation (RAG)
- Direct, structured data access
- Faster, more precise queries
- Better for domain-specific tasks
- Seamless LLM integration

Flow of Instructions

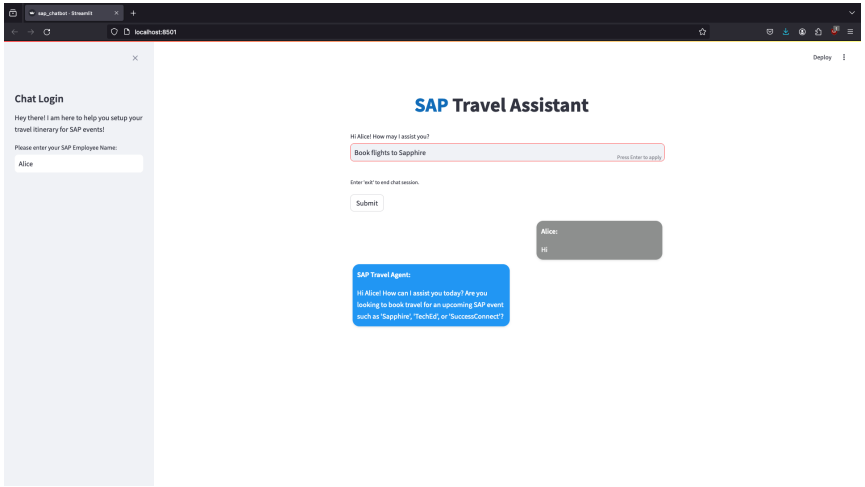
- The flow is a continuous loop:

    1. Start `main()` → Initialize state → Display UI
    2. User enters name → Validate → Update state → Set up chat or show error
    3. User enters query → `submit()` function → Parse query → Call `handle_chat_tmp2`
    4. `handle_chat_tmp2` → Use agents → Get data → Format → Get GPT response
    5. Update chat history → Display new messages → Wait for next input

Future Enhancements

- Integrate with actual SAP APIs and databases.
- Add more agents (e.g., Booking Agent, Payment Agent).
- Add multi-language support
- Integrate with calendar APIs for scheduling.
- Implement voice-based interactions (or multi-modal)
- Add personalized recommendations based on user preferences.
- Integrate with external services (e.g., weather, local events).
- Implement end-to-end encryption for security.
- Add unit tests and integration tests.

Live Demo - See It in Action



- Query: "Book me a flight to Sapphire"
- Watch agents collaborate in real-time
- See flight and hotel options
- Experience natural conversation