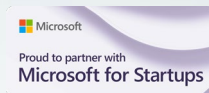


# How I Rebuilt ResumeGuru.IO into a Multi-Agent AI system - and How You Can Too

By Solomon Tsao

Founder of ResumeGuru.IO



# What We'll Cover Today

**The “Before” :** The Pains of a Rule-Based System

**The "Aha!" Moment:** Shifting to a Multi-Agent Architecture

**Our AI Toolkit:** A Look at LangChain, AutoGen, RAG, and MCP

**The "Before -After -Future":** A history of our Interview Agent

**Lessons Learned:** Key Takeaways from Deploying to Production

**Demo:** Build Your Own Agent & Live Demo



# Life with a Rule -Based System

- We even hardcoded pushing the user's resume and a job description to the LLM provider. (Dec / 2022)
- The limitations:
  - **Brittle and Inflexible:** "If-then-else" logic couldn't handle the variety of human expression.
  - **Hard to Maintain:** Every new rule or skill required complex, manual coding.
  - **Limited Scope:** It was a "dumb" assistant, not a true career partner.





# The "Aha!" Moment: The Rise of AI Agents

1

Sep 2023, the release of AutoGen 0.1 framework from Microsoft.

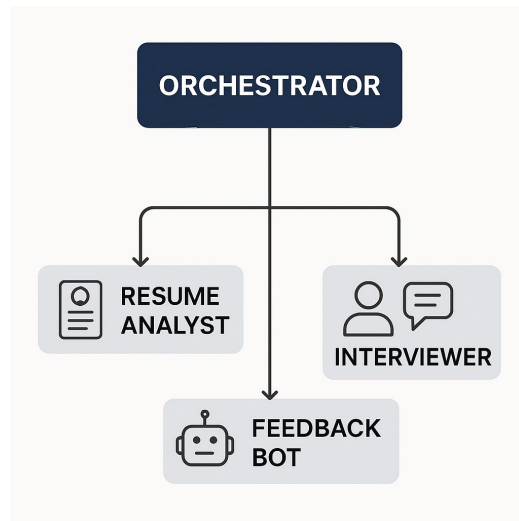
**Two-agent chat, Sequential chat, Group Chat, Nested Chat**

2

"Specialized AI workers with specific jobs."

[AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation](#)

[Microsoft AutoGen](#)





## LangChain:

The glue for AI agents

LangChain connects large language models (LLMs) to various data sources and tools. Chains, memory, and tool-use are its core features.

## AutoGen

The framework for agent conversations

AutoGen orchestrates dynamic, conversable agents working collaboratively.

# Why AutoGen? A Look at the Agent Landscape



We chose **AutoGen** for its strength in creating dynamic, automated conversations that best simulate an unpredictable job interview, combined with **LangChain's** robust tools for connecting to our data.



Framework	Core Concept	Control Flow	Best For...
<b>AutoGen</b>	Flexible multi-agent conversations	Automated & dynamic conversation between agents	Research, complex problem-solving, and allowing agents to determine the best path.
<b>LangGraph</b>	State machines as a graph	Explicitly defined cycles and edges. Full control.	Complex, stateful workflows that require loops, branching, and high predictability.
<b>CrewAI</b>	Collaborative "Crews" with roles	Process-oriented; agents with roles follow a defined process.	Role-playing tasks, creative content generation, and mimicking human team structures.

# The "After" Picture: The Multi-Agent Redesign



- **User Proxy Agent:** Represents the human user.
- **Recruiter Agent:** Simulates a recruiter by asking interview questions.
- **Job Description Analyst Agent:** Parses and understands the target job.
- **Resume Analyst Agent:** Critiques the resume against the job description.
- **Feedback Agent:** Provides constructive feedback on the user's answers.



Open-sourced today

# The "After" Picture: The Multi-Agent Example




```
self.interviewAgent = autogen.AssistantAgent(  
    name="InterviewerAgent",  
    llm_config=llm_config_assistant,  
    system_message="You are the Interviewer Agent and your name is Hannah. You work for ResumeGuru.IO. "  
    "You ask relevant technical and behavioral questions based on the job description and user input. "  
    "Search database first and list the newest 5 jobs with company name and job title. User can decide which one to start the interview process."  
    "If no job description is found, ask the user to provide the job description."  
    "Each time only ask one question. Do not return any special characters like hashtags or brackets. ",  
    code_execution_config={"use_docker": False}  
)
```

```
self.dbSearchAgent = autogen.AssistantAgent(  
    name="DatabaseSearchAgent",  
    llm_config=llm_config_assistant,  
    function_map={  
        "search_db_job_description": self.search_db_job_description  
    },  
    tools=[self.search_db_job_description],  
    reflect_on_tool_use=False,  
    system_message="Your job is to search the database for all of the job description based on the user id :"+str(self.user_id)+" and return the result to the user.",  
)  
  
self.feedbackAgent = autogen.AssistantAgent(  
    name="FeedbackAgent",  
    llm_config=llm_config_assistant,  
    system_message="You provide real-time feedback on the candidate's answers, analyzing them for completeness, relevance, and clarity.",  
)  
  
self.critic = autogen.AssistantAgent(  
    name="critic",  
    llm_config=llm_config_assistant,  
    system_message="""You are a helpful assistant, you should validate the ideas from the interview agent, once done return the idea to the user""")  
)
```



# The "After" Picture: The Multi-Agent Example





**1.0%**  
Content Score

**1.0%**  
Pronunciation Score

**Score Detail**

**Content** Pronunciation

**Grammar Score** 1%

Proficiency of the correctness in using grammar. Grammatical errors are jointly evaluated by incorporating the level of proper grammar usage with the lexical.

**Topic Score** 1%

Proficiency in lexical usage, which is evaluated by speaker's effective usage of words, on how appropriate is the word used with its context to express an idea.

**Vocabulary Score** 1%

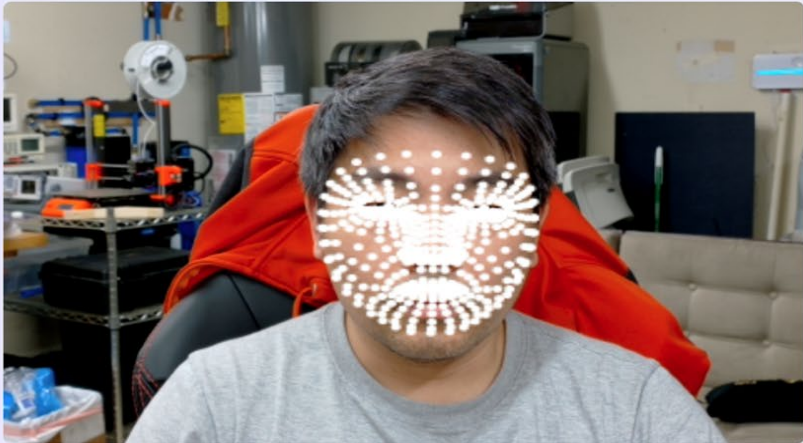
Proficiency in lexical usage, which is evaluated by speaker's effective usage of words, on how appropriate is the word used with its context to express an idea.

Solomon Tsao

Perfect!

Perfect!

Perfect!



Display recognition result? ☒

**End Session**

**Recommended Answer**

**Quick Reply Prompts**

Where should I start?

I don't know how to answer, please guide me?

I dont understand the question, can you explain it?

Next question please

I am not sure about the answer, can you help me?

More Job Options. Yes No

<https://yottatrend.com/ResumeGuruIODemoVideo>

# The "Future" Picture : the agentic workflow

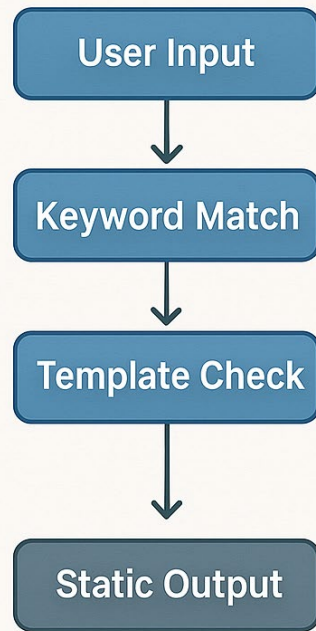
## Agentic Agent:

An agentic agent is an AI entity capable of acting autonomously, making decisions, and taking actions to achieve specific goals with minimal human intervention.

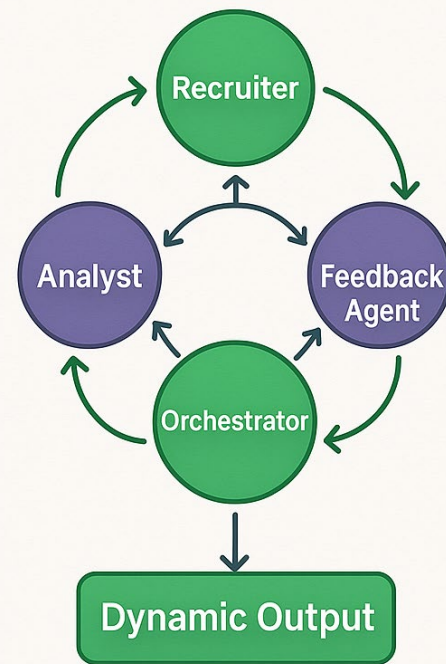
## The New Way (Agentic):

A request kicks off a dynamic conversation. Agents can talk to each other, delegate tasks, and even ask for help. It's flexible and intelligent. Example : a panel interview.

## Old Way



## New Way



# The "Future" Picture : give agents knowledge

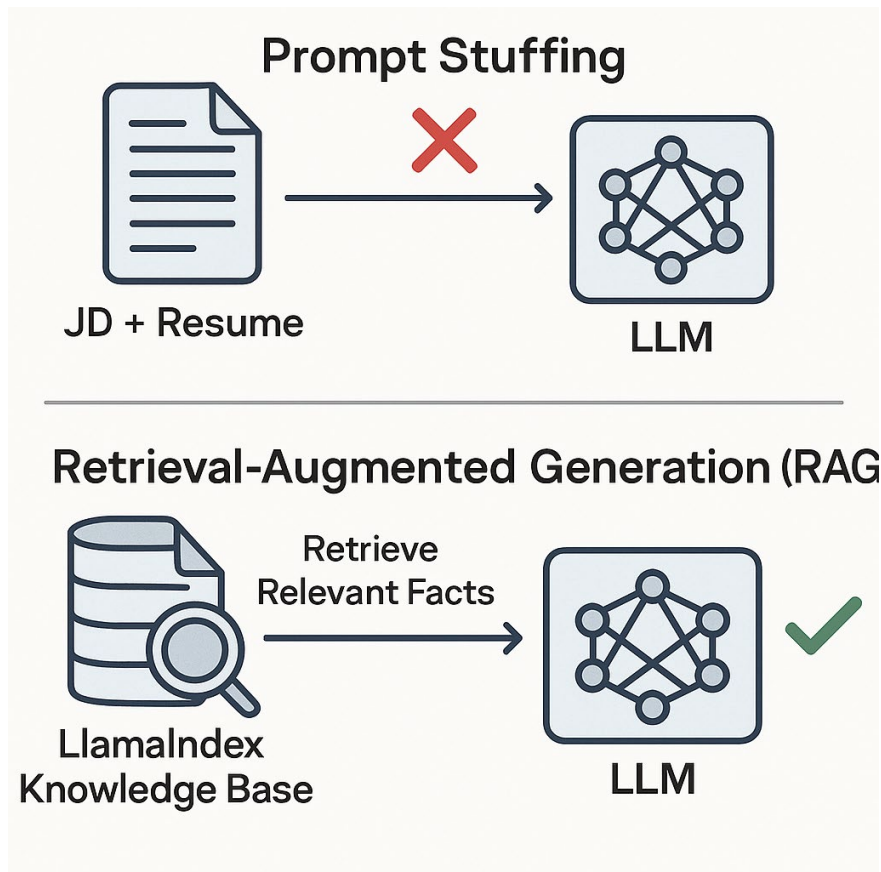


## The Old Way (Prompt Stuffing):

- Shove the entire resume and job description into the prompt with every query.
- **Problems:** Inefficient, costly, and context gets lost in the noise.

## The Smart Way (Retrieval - Augmented Generation - RAG):

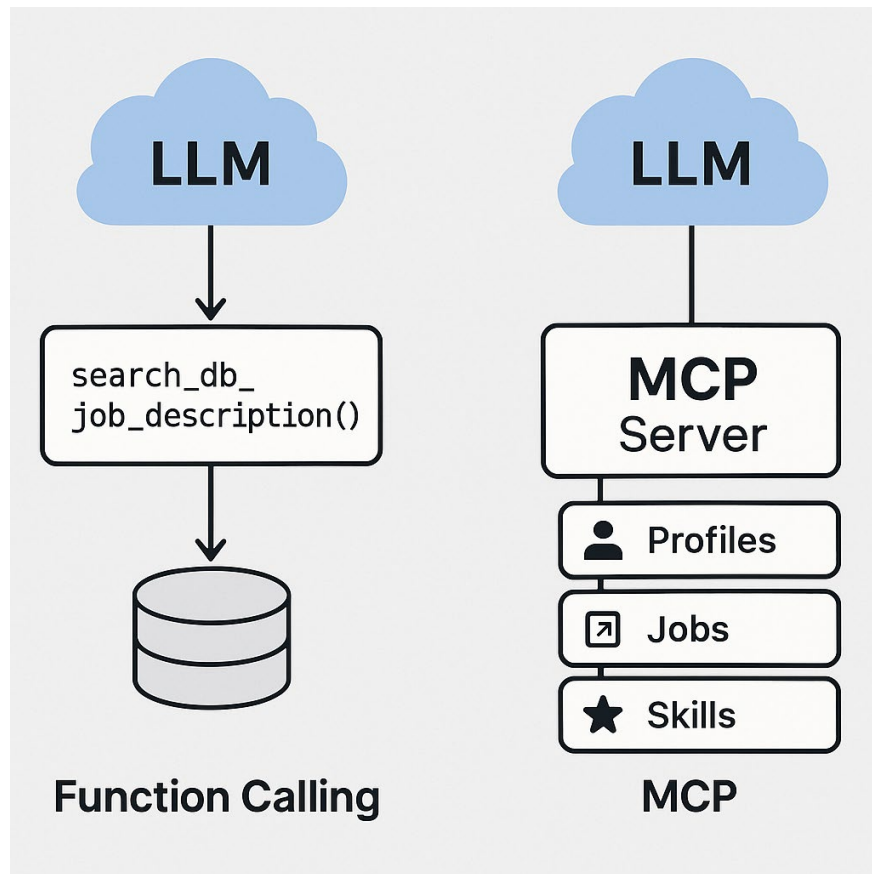
1. **Index:** LlamaIndex to read the resume and job description once, creating a structured, searchable **Knowledge Base**.
2. **Retrieve:** When the Recruiter Agent needs info, it queries this Knowledge Base to find the *exact relevant snippets*.
3. **Augment:** It adds these small, relevant snippets to its prompt for the LLM.



# The "Future" Picture : function calling to MCP

## The Future (Model - Context - Protocol - MCP):

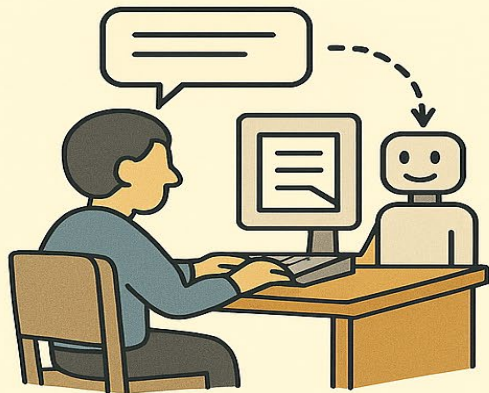
- **What it is:** Exposing a standardized, browsable "context" or resource, like a UserProfile API. The LLM can then make structured queries against it.
- **Analogy:** Giving a person a full, searchable company directory. They can find the right person and the right information on their own.
- **Our Vision:** A public ResumeGuru MCP server could allow other agents to securely query anonymized job market data, creating a powerful developer ecosystem.



# Lessons Learned from Production :



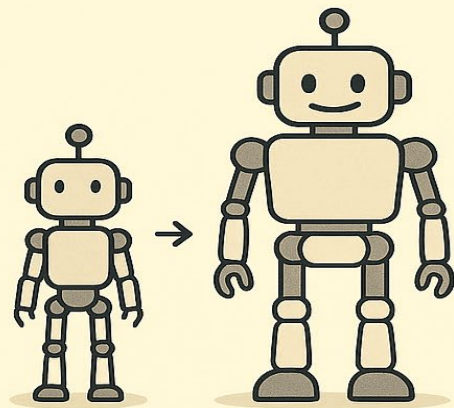
**Lesson 1**  
**Prompt**  
**Engineering**  
**is 80% of the**  
**Work**



**Lesson 2**  
**System**  
**Architecture**  
**is Key**



**Start Small**  
**Then Scale**



## Build your own agent - Vibe Coding

"vibe coding" is about leveraging the advanced capabilities of LLMs and agentic AI to democratize software development. A good design prompt is the most critical part to start.

Tools : Claude code, Cursor, Windsurf, Bolt, Lovable, Gemini CLI



# Start with a structured prompt :

A decorative horizontal bar with a teal segment on the left and an orange segment on the right.

## Project Overview

## Technology Stack

## Architecture Design

### Core Components

### AI Agent Design

#### Agent Personality & Strategy

##### Prompt Engineering Strategy  
...

System: Your system prompt  
...

### User Interface Design

### Data Flow Architecture

#### AI Integration Flow





"Describe the agent you want to build" +  
"design prompt template"  
= Your tailored design prompt for vibe coding

Design\_prompt\_template =>

[https://drive.google.com/file/d/1rjnTzL9vZWxhDLf0iW0YxET4OYoSFoGF/view?usp=drive\\_link](https://drive.google.com/file/d/1rjnTzL9vZWxhDLf0iW0YxET4OYoSFoGF/view?usp=drive_link)

Design\_prompt\_example =>

[https://drive.google.com/file/d/1gXxUNmjAgpEf\\_ABH2f9\\_EgxKJuHgDyRQ/view?usp=sharing](https://drive.google.com/file/d/1gXxUNmjAgpEf_ABH2f9_EgxKJuHgDyRQ/view?usp=sharing)





## Links for all of the code :

**ChaseMind** - <https://github.com/solomon2773/chess-mind-unleashed/tree/main>

**Coffee Concierge design sample** - <https://yottatrend.com/joSBh>

[ResumeGuru.IO](https://github.com/solomon2773/ResumeGuru.IO) (Source code) - <https://github.com/solomon2773/ResumeGuru.IO>



# Thank you.

LinkedIn : <https://www.linkedin.com/in/solomon-tsao/>

Email : [solomon.tsao@gmail.com](mailto:solomon.tsao@gmail.com)

