Designing conversational agent(ic) systems:
Concepts, Capabilities, and Considerations

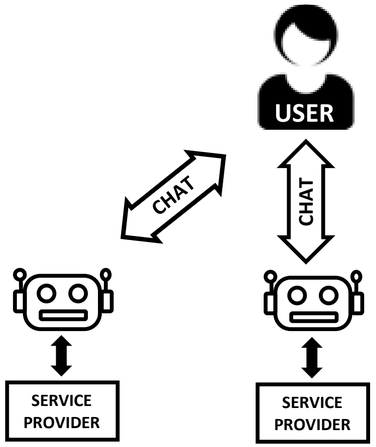
Heloisa Candello
Michelle Brachman
Amanda da Silveira

IBM Research

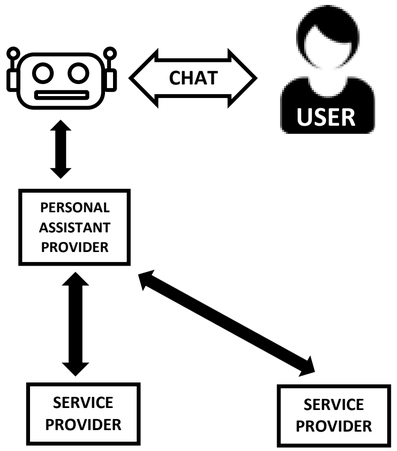
Agenda

1. Introduction: From Rule-Based Bots to Conversational Agent(ic)
2. Defining Agentic systems
3. Designing conversational agentic systems
4. Risks, challenges and mitigation approaches
5. Frameworks for Agentic Systems
6. Step-by-step applied Framework
7. Future directions

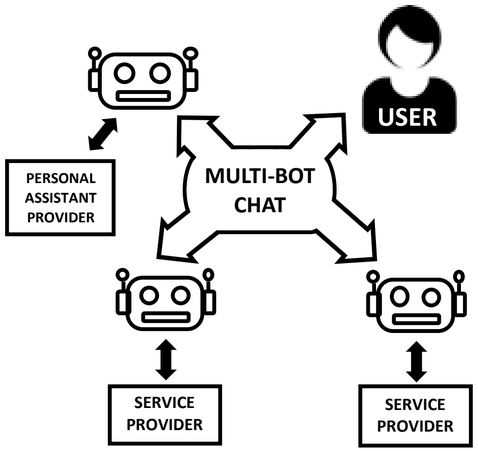
From Rule-Based Bots to Conversational Agent(ic)



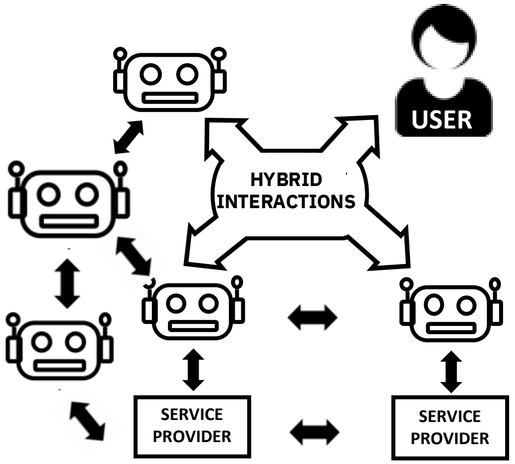
EXPERT AGENTS



PERSONAL ASSISTANTS



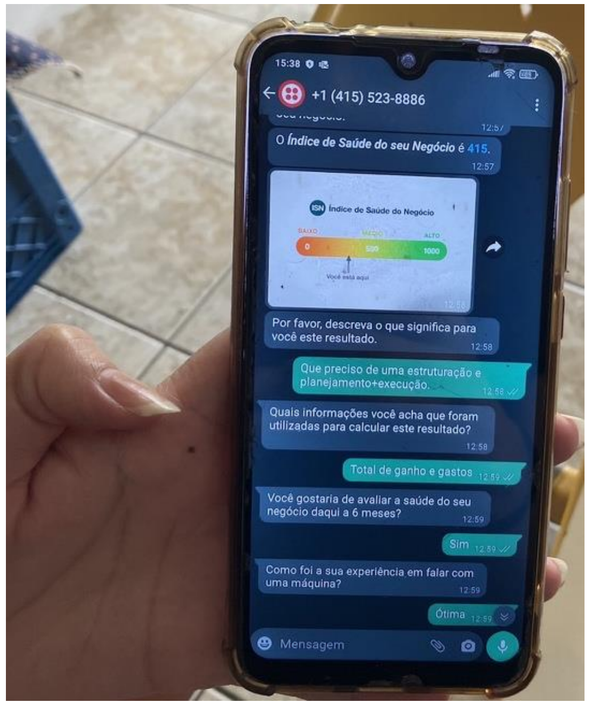
TEAMS OF EXPERT AGENTS

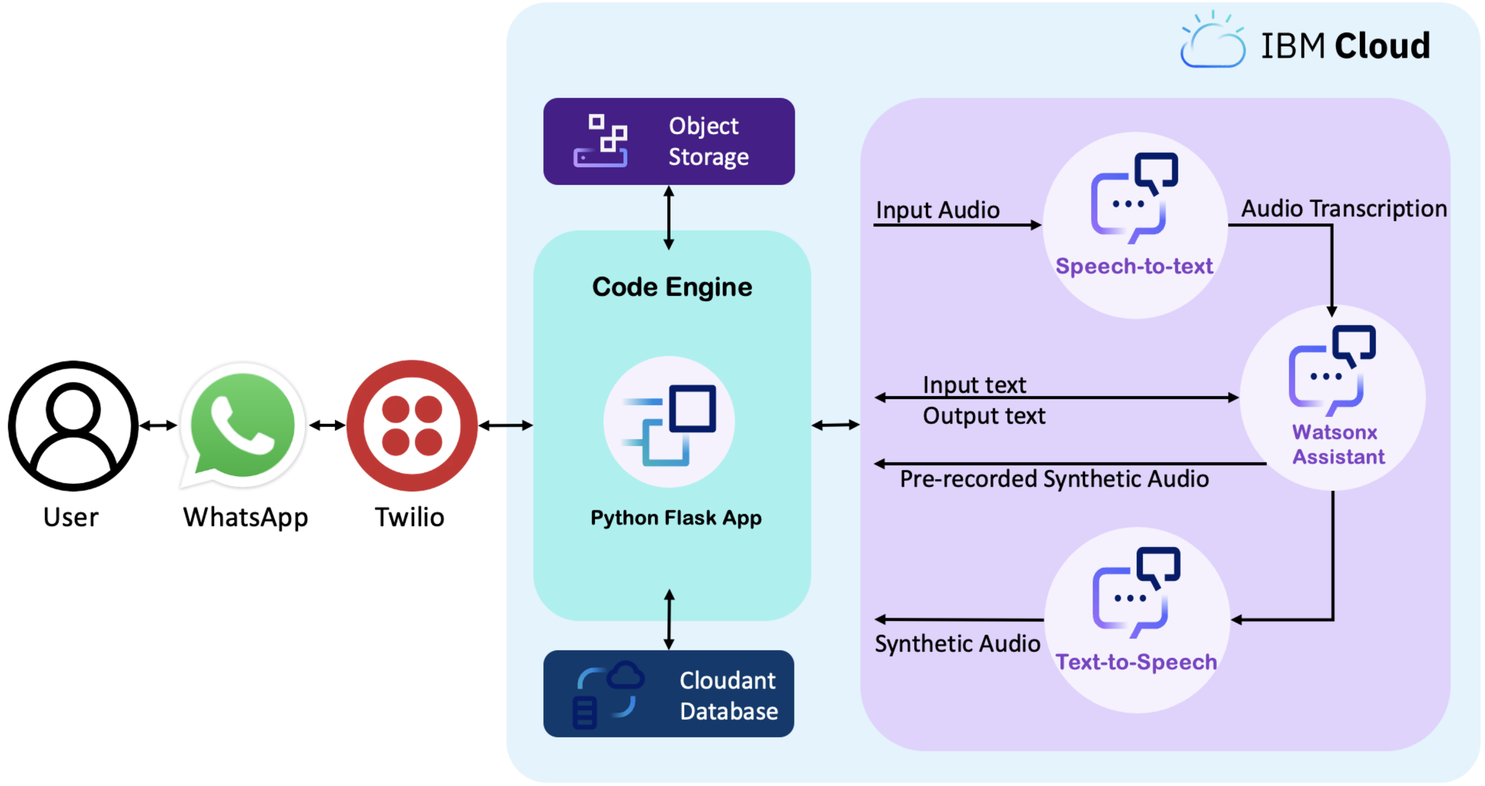


AGENTIC SYSTEMS

1970s–2000s: Finite-state and rule-based systems – text and voice-based chatbots
2010s: Neural dialogue systems – learning-based dialogue
2020s: Generative and agentic AI – reason, plan and take actions

Single-bots



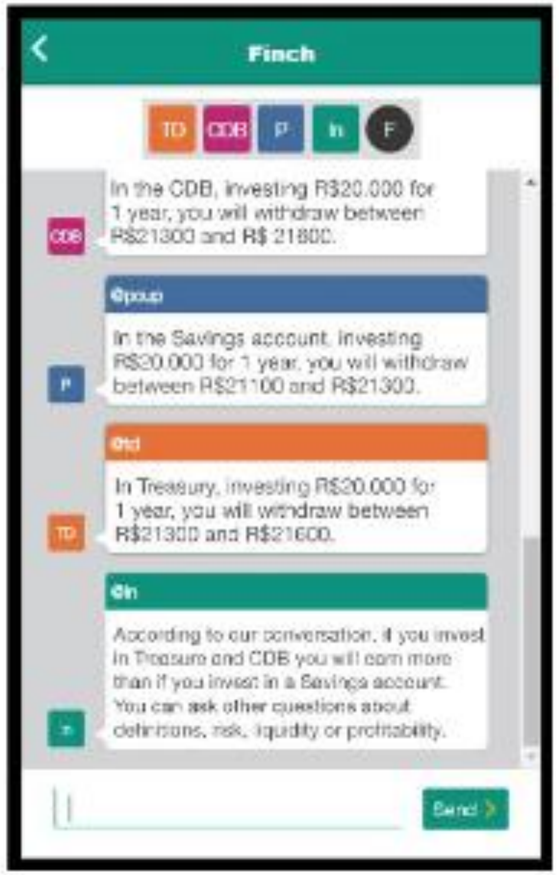


System architecture

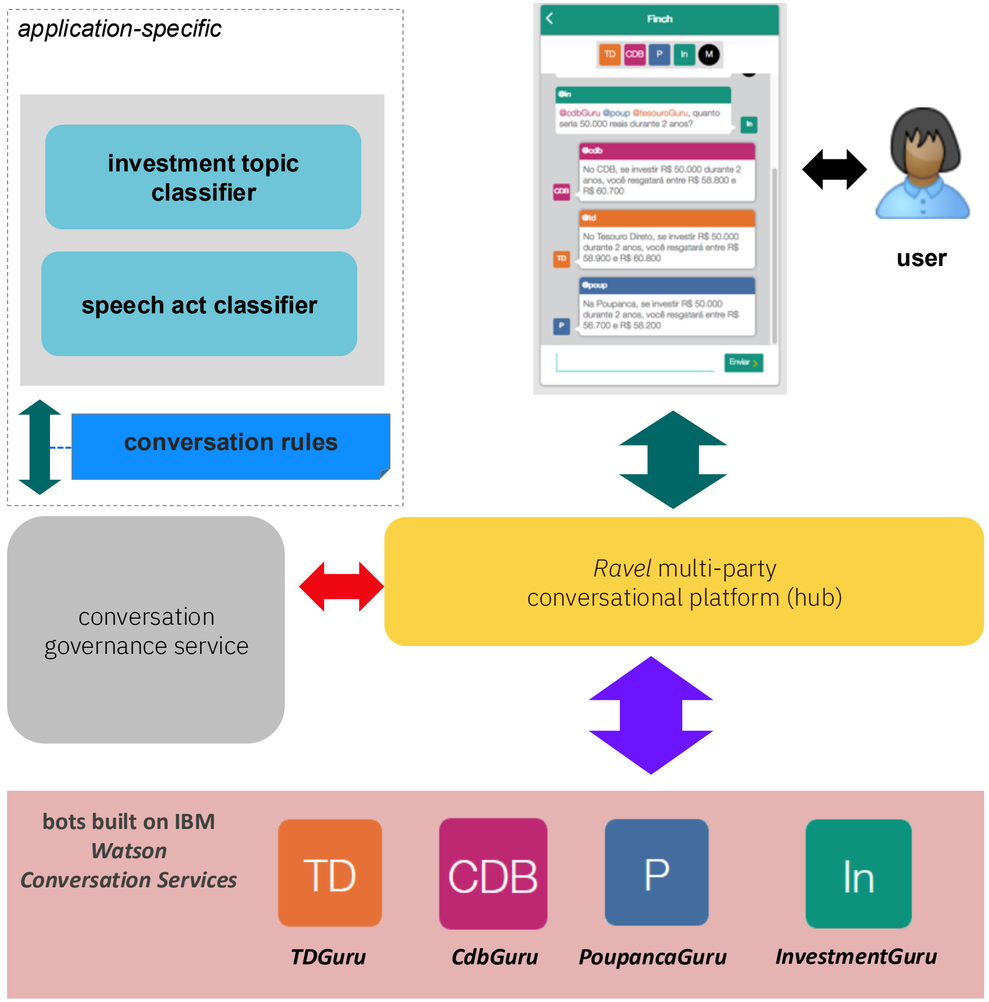
Open-source :
https://ibm.github.io/customized-voice-text-bot-for-whatsapp-telegram/

[CHI 22 , FAccT22 , CUI 22, CUI23; HCI@NeurIps21, AAAI, IMX24]

Multi-bots



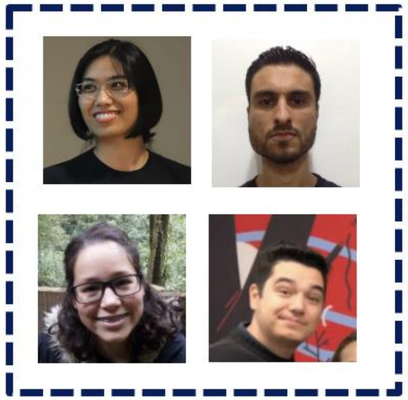
from Q&A dialog to multiparty chat

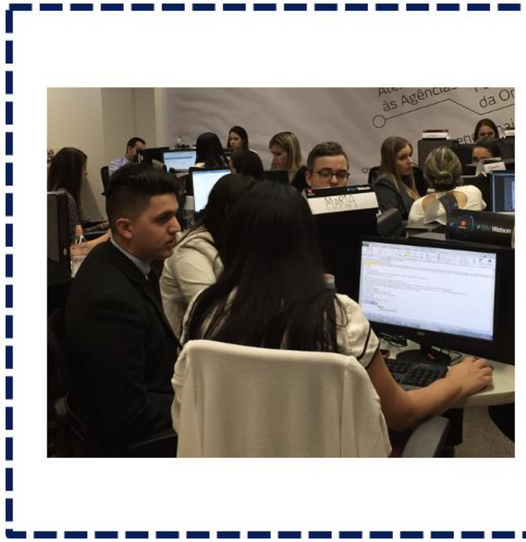


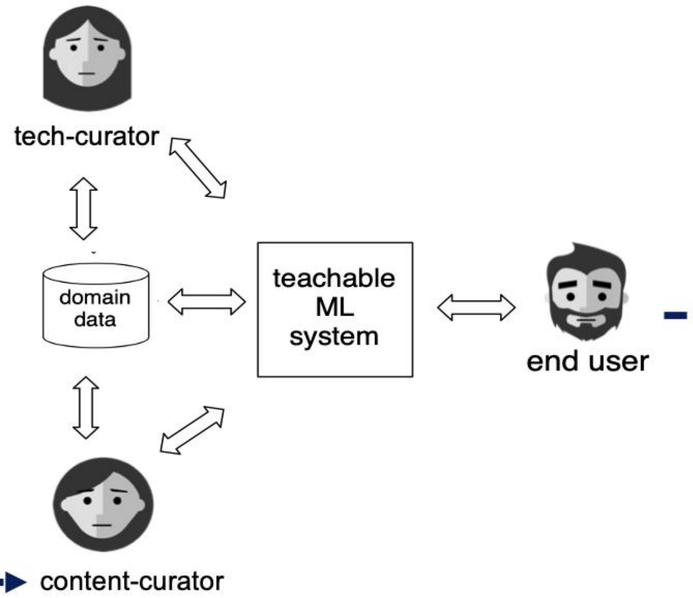
Social conversation and turn-taking defined by application developer using **orchestration rules**.

[CUI2021, CHI2019, CHI2020, DeepDIal 2018]

Workflow – teaching conversational systems







Machine Teaching (MT) is the research discipline that aims to make people more productive in the creation, and maintenance of machine learning systems. It aims to expand to **non-ML specialists the ability to create ML systems**, making the teaching process more natural.

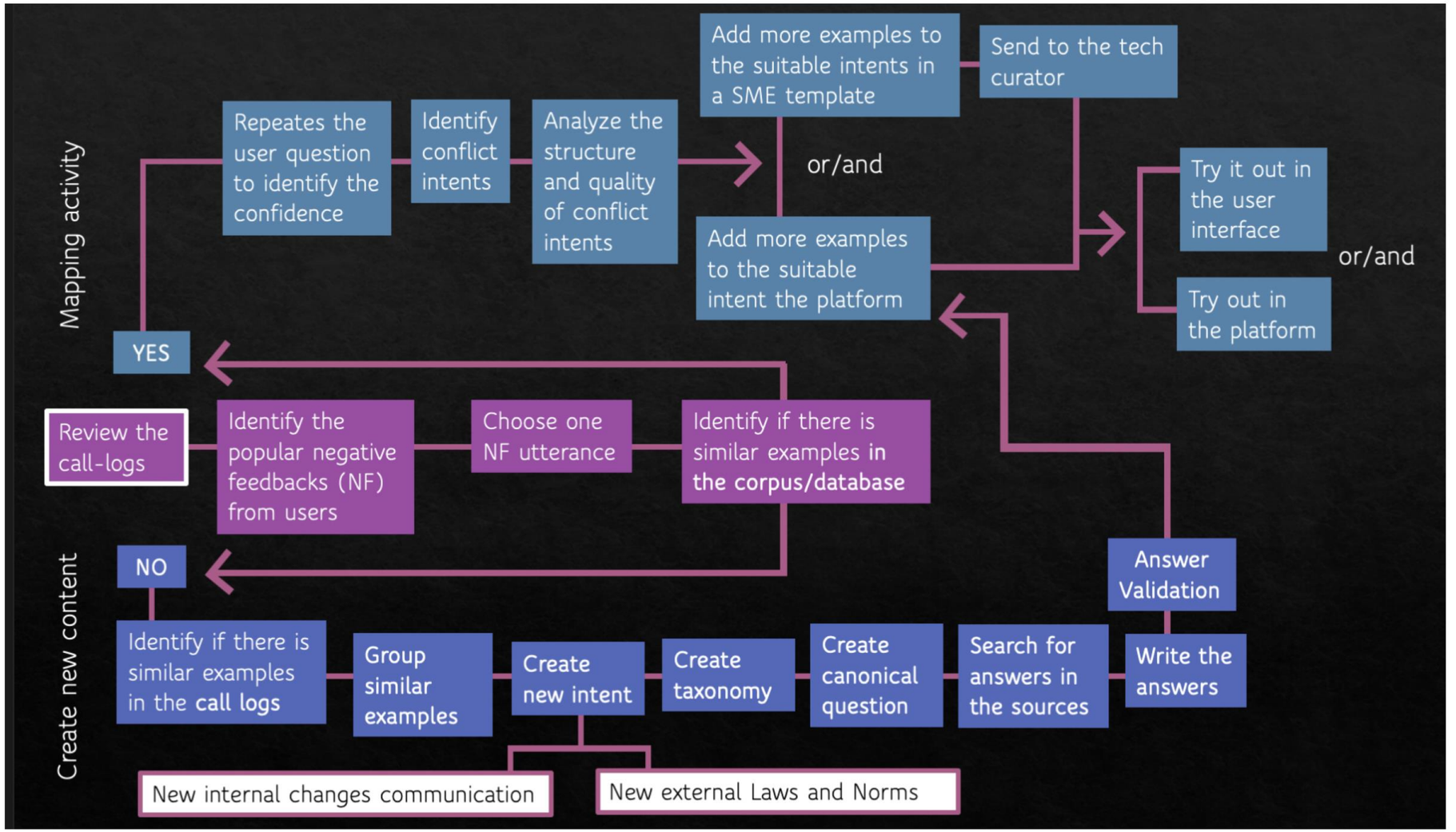
[ACM CSCW2022]

Workflow

Negative Feedback

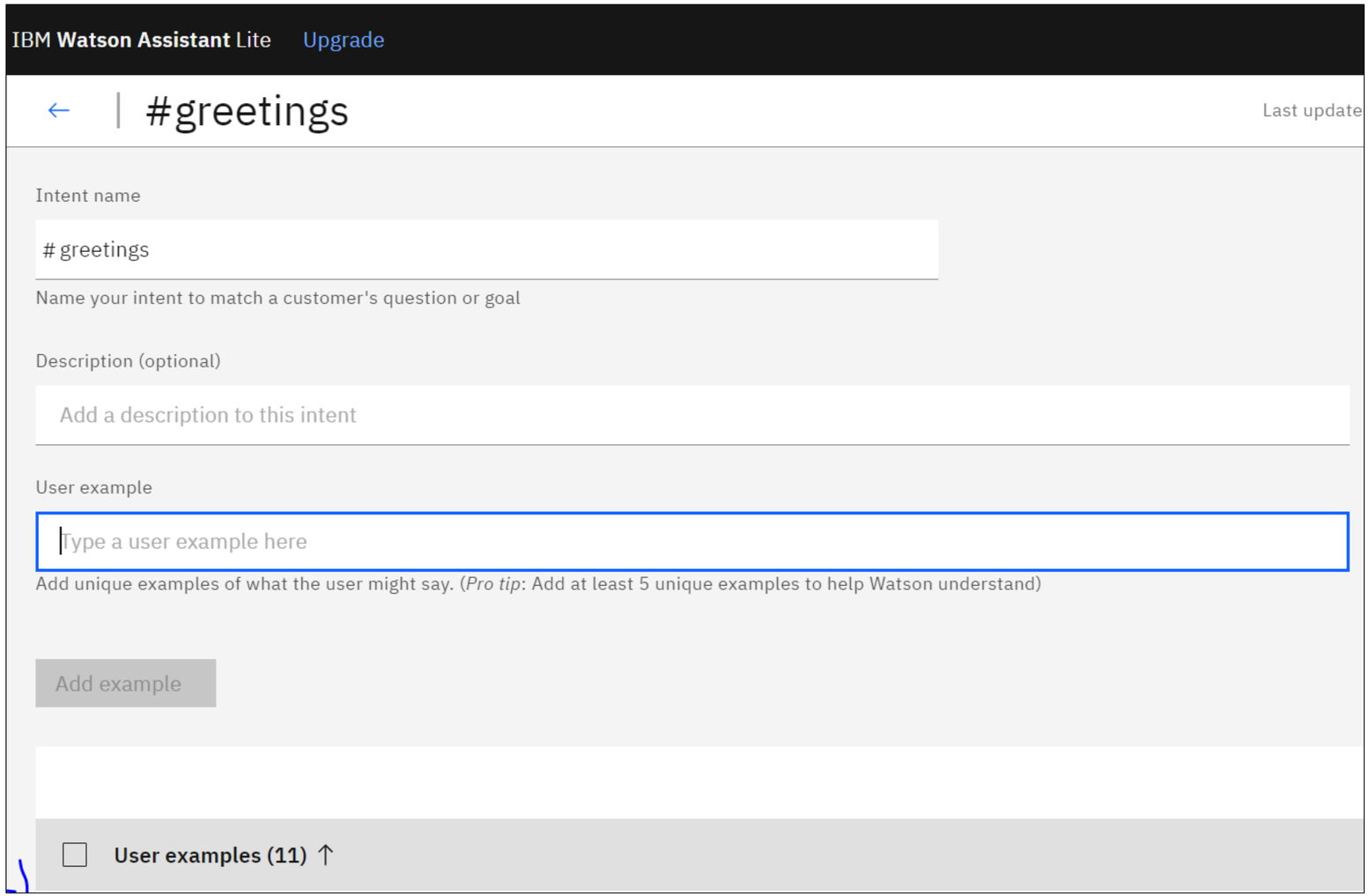
User: How do I make payments?
Bot: Sorry, I don't know how to answer it yet.

Content curators
Domain experts

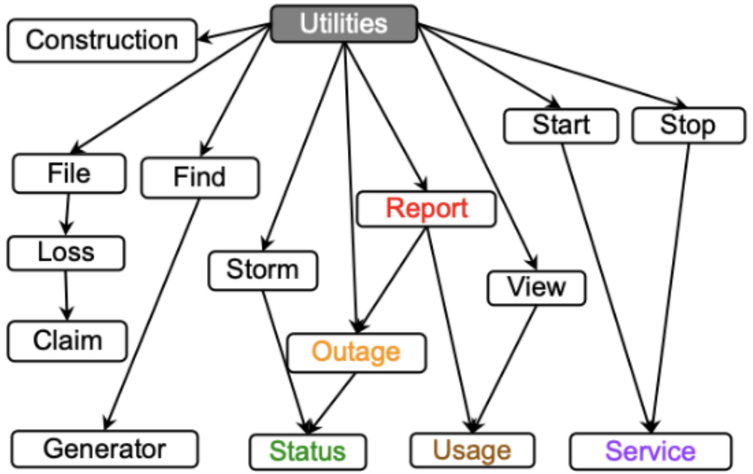


Flowchart illustrating a case where content curation articulation of work takes place

Platforms for CUIs



Watson Assistant platform - based on user intent examples



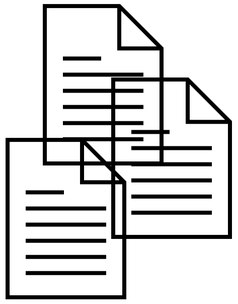
The intent proto-taxonomy associated to the utilities-related intents

"Using meta-knowledge mined from identifiers to improve intent recognition in conversational systems." *ACL*2021.

IBM **Research**

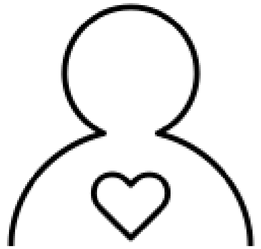
Agentic systems
AI for Thematic Analysis

Qualitative Data
interviews, open-ended survey questions



Analyst
user experience researcher

Customer
product manager



Types of thematic analysis

Inductive
Themes emerge from the data

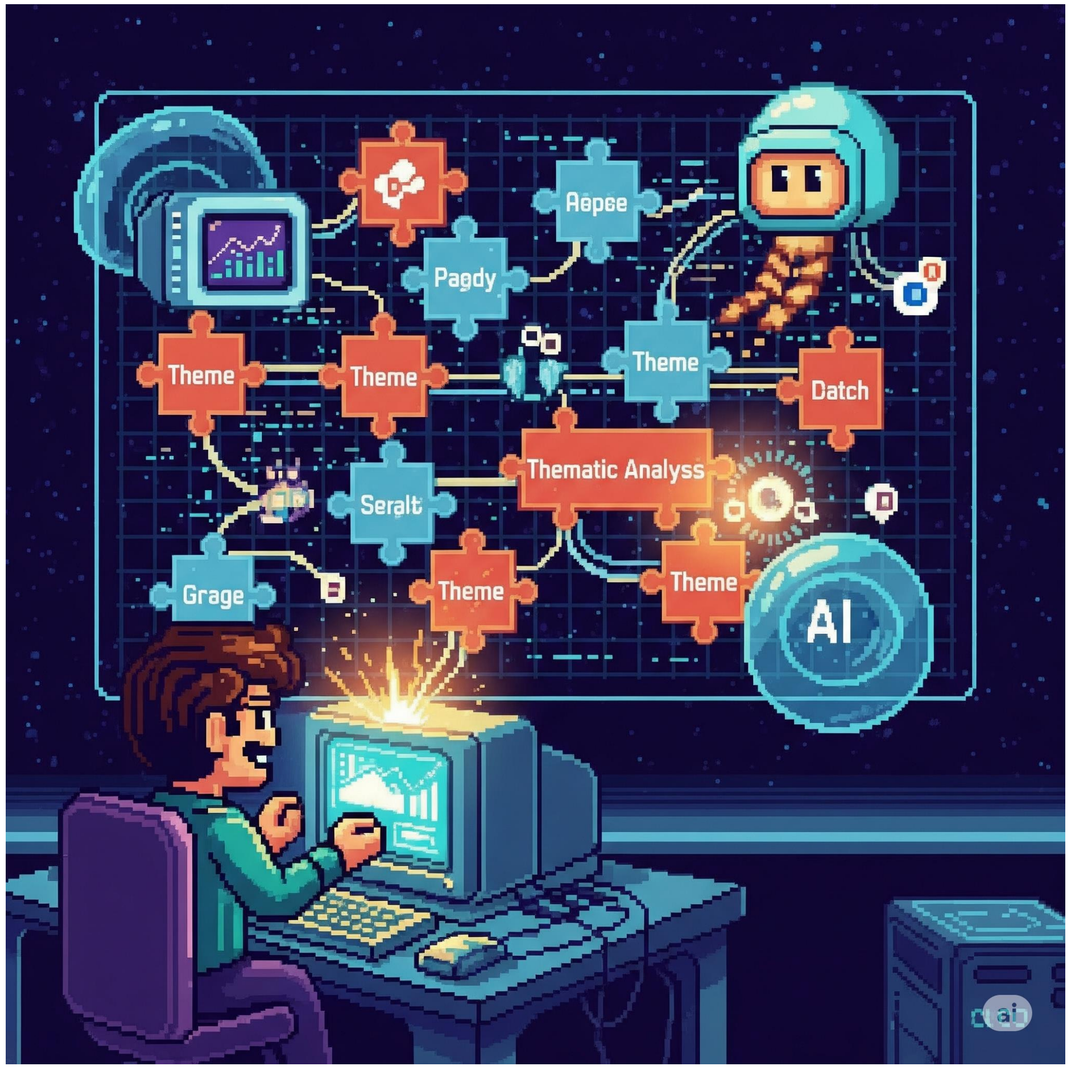
Reflexive
Themes emerge iteratively through an ongoing dialogue between the data and the researcher

Deductive
Pre-existing theoretical framework or hypothesis guides the identification of themes

AI to the rescue?

Can we make thematic analysis feel more like “vibe coding” while **preserving its integrity**?

- How can we help analysts & customers trust AI-assisted insights?
- How can we design new workflows with high levels of human and AI agency?
- How can we ensure (novice) analysts develop their skills vs. outsourcing everything to AI?



Traditional CUIs

- Predefine responses
- Responses displayed to end users after human curation
- Answering according confidence threshold
- False positives existed and were defined by humans
- Limited possibilities and tied to prior training

What’s new with LLM?

- LLMs enable open-ended responses
- Can hold memory, plan, use tools
- Support complex, ambiguous tasks
- Lower barrier to entry for builders
- Amplified AI risks

Defining Agentic Systems

.

Agentic Systems Overview

There are many ways to define agentic systems.

*"An AI agent is a software entity that employs AI techniques and **has agency** to act in its environment based on set **goals**, which means it **can decide** which actions to perform and has the ability to execute them."*

https://www.ibm.com/downloads/documents/us-en/1227c12efb38b2b3

*"AI agents are **autonomous technologies**, designed to perform tasks, make decisions, and interact with their environment to achieve specific predefined goals, with **minimal human intervention**."* *(Hughes, 2025)*

*"AI agents aim to **solve complex tasks** by combining text-based reasoning with external tool calls"*

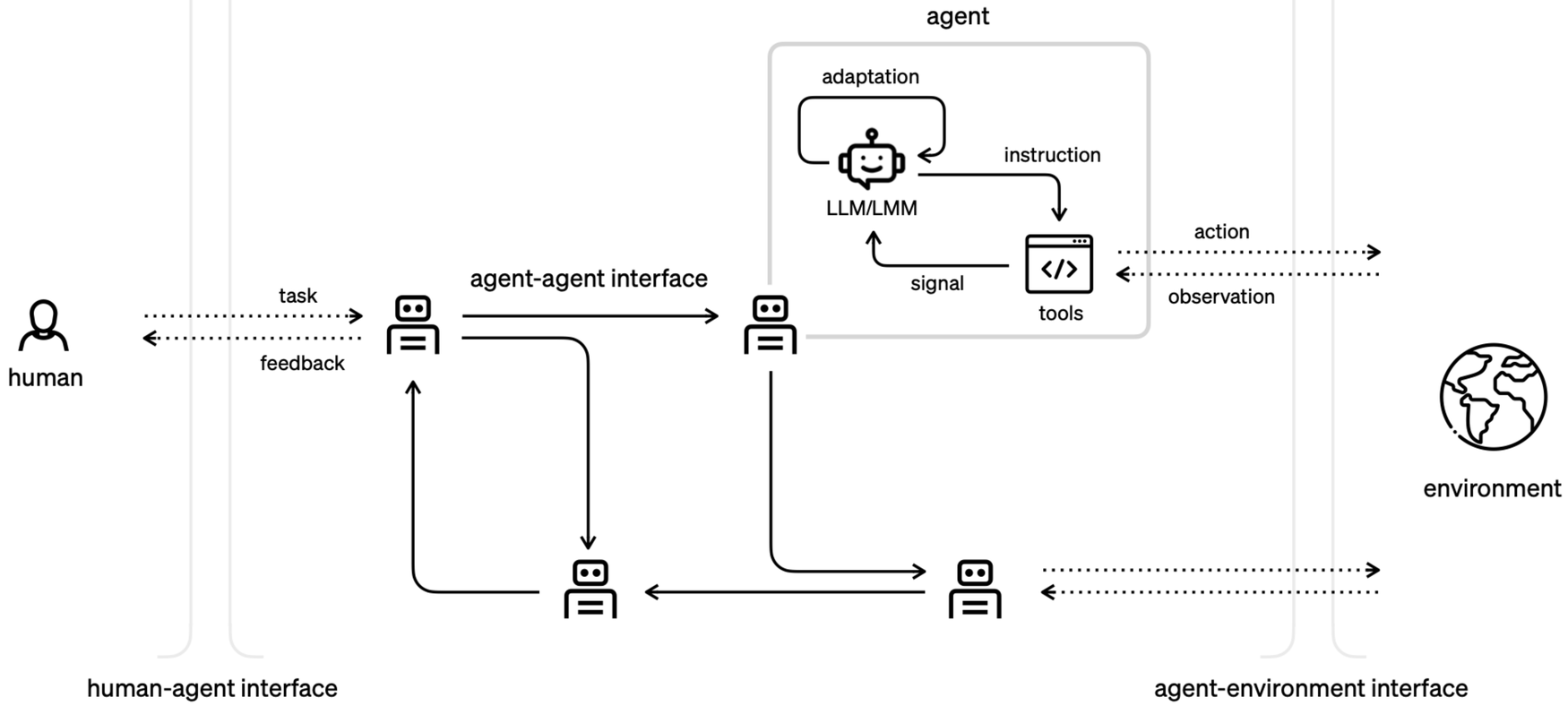
(Debenedetti et al. 2024

*"The advent of AI agents, capable of **autonomous perception and action,** further compounds this issue by exposing the **limitations of traditional human centered research approaches.**"* *(Zhang et al. 2024)*

*"An agentic AI system […] **is a collection of agents interacting with humans and the environment** with the objective of fulfilling specified goals. Practically, an agent is an LLM or large multimodal model (LMM) with access to tools — specialized components/functionalities like APIs, external services, computational resources, or domain-specific software — that allow it to perform specific operations in the environment."* (Miehling, 2025)

IBM **Research**

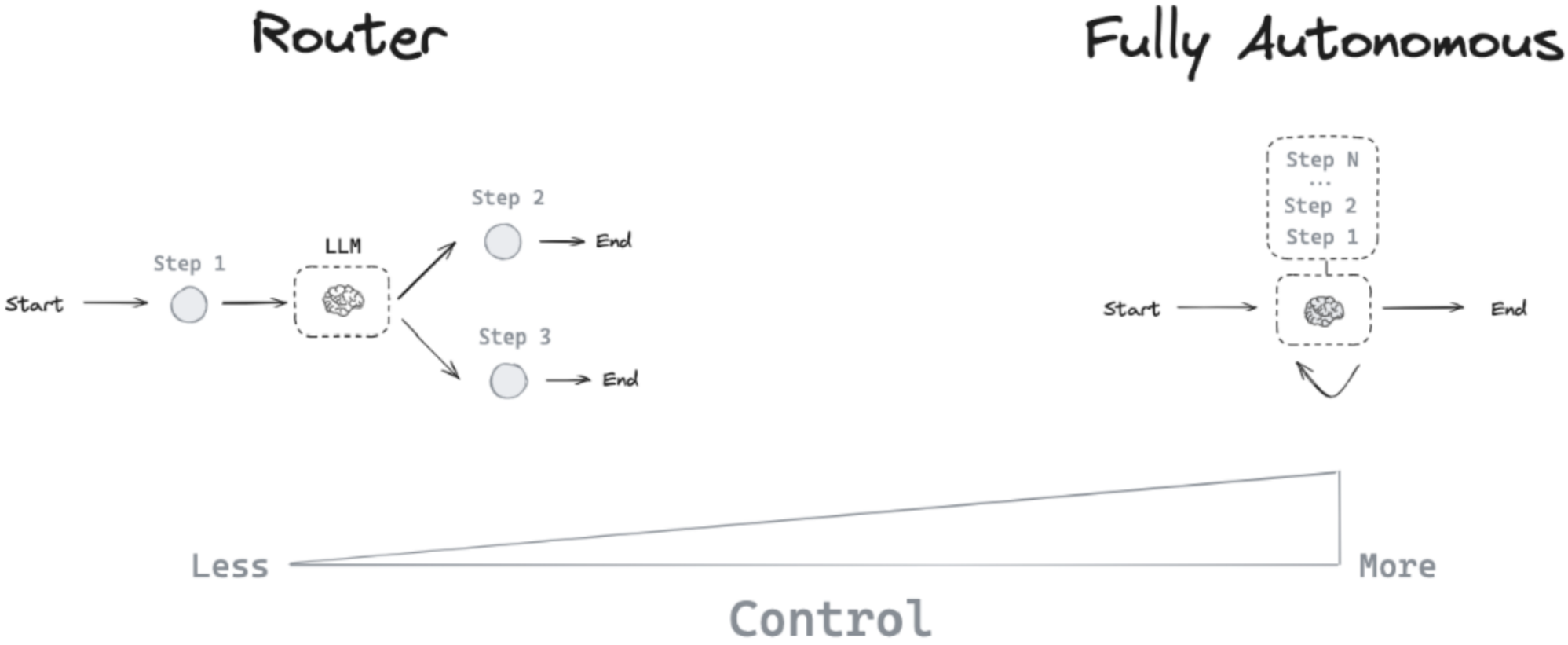
Agentic Systems Overview



"An agentic AI system […] ***is a collection of agents interacting with humans and the environment*** *with the objective of fulfilling specified goals. Practically, an agent is an LLM or large multimodal model (LMM) with access to tools — specialized components/functionalities like APIs, external services, computational resources, or domain-specific software — that allow it to perform specific operations in the environment."*

Miehling, E., Ramamurthy, K. N., Varshney, K. R., Riemer, M., Bouneffouf, D., Richards, J. T., ... & Geyer, W. (2025). Agentic AI Needs a Systems Theory. *arXiv preprint arXiv:2503.00237*.

Agentic Systems Overview



“an agent is a system that uses an LLM to decide the control flow of an application”

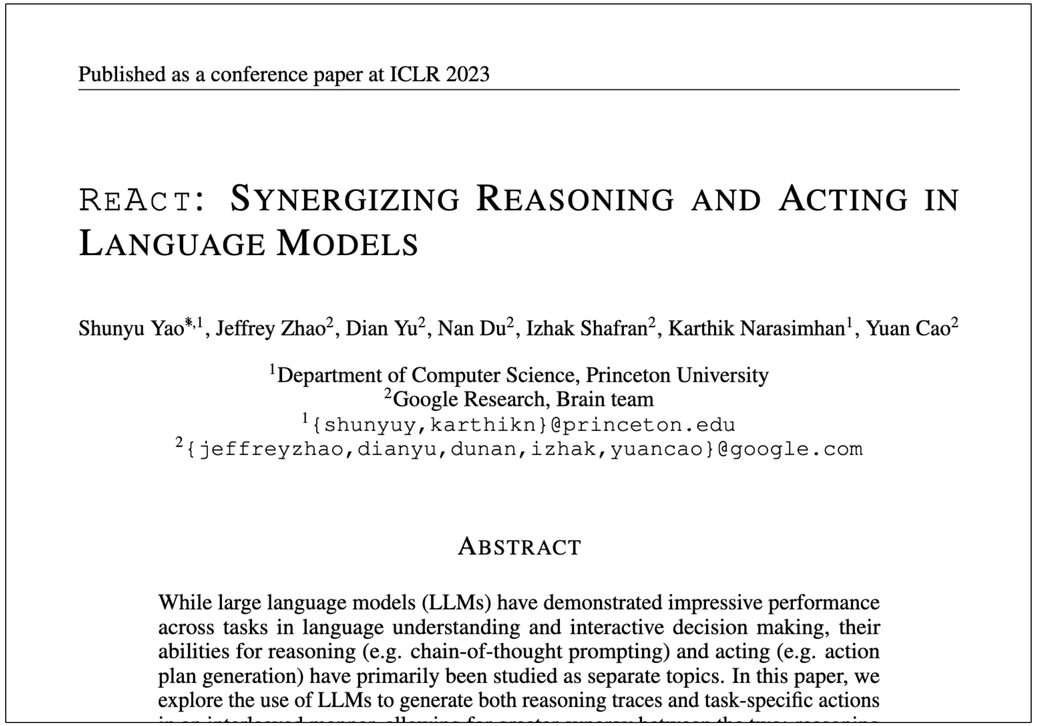
1From https://langchain-ai.github.io/langgraph/concepts/agentic_concepts/

Agentic patterns: ReAct

Traditional vs **React**

- Scripted
- Reactive

- Decision-making to satisfy the goal
- Coordinate with other actors
- Adapt



Published as a conference paper at ICLR 2023

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao*,[1], Jeffrey Zhao[2], Dian Yu[2], Nan Du[2], Izhak Shafran[2], Karthik Narasimhan[1], Yuan Cao[2]

[1]Department of Computer Science, Princeton University
[2]Google Research, Brain team
[1]{shunyuy,karthikn}@princeton.edu
[2]{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

ABSTRACT

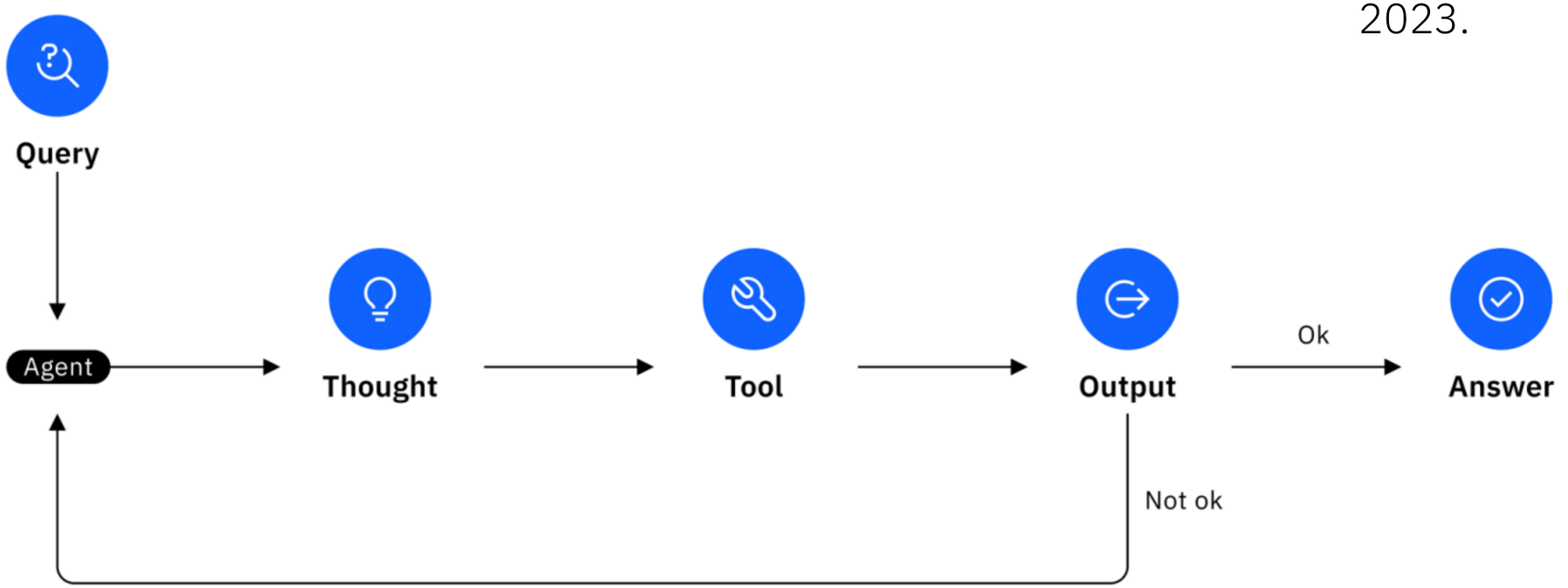
While large language models (LLMs) have demonstrated impressive performance across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation) have primarily been studied as separate topics. In this paper, we explore the use of LLMs to generate both reasoning traces and task-specific actions

Yao, Shunyu, et al. "React: Synergizing reasoning and acting in language models." *International Conference on Learning Representations (ICLR)*. 2023.

ReAct (Reasoning and Acting) is a framework that combines chain-of-thought reasoning (comparable to “thinks out loud”) with the ability to interact with external tools, enabling AI agents to solve complex tasks more effectively. It allows agents to dynamically plan, act, and refine their approach based on the results of their actions, mimicking human problem-solving more closely.

Agentic patterns: ReAct

– Yao, Shunyu, et al. "React: Synergizing reasoning and acting in language models." *International Conference on Learning Representations (ICLR)*. 2023.



ReAct (Reasoning and Acting) is a framework that combines chain-of-thought reasoning with the ability to interact with external tools, enabling AI agents to solve complex tasks more effectively. It allows agents to dynamically plan, act, and refine their approach based on the results of their actions, mimicking human problem-solving more closely.

Components of Agentic Systems

Agents

Components that perform actions/communicate with each other or an external entity
May have roles/descriptions
Have ways of communicating with each other
Can be pre-built or custom

Model or models

Large language models are used for the planning process as well as other aspects of agentic systems.

Planning/reasoning/ decision engine

How the system decides what to do next or the sequence of steps to take.

Memory/State

Information that is stored and used across multiple steps or interactions. It can be short-term or long-term.

Tools

Tools are capabilities that the system can use to accomplish parts of the goal, such as a calculator, search capability, or a call to an LLM.

Observability/logging

Ability to observe or keep track of what the system is doing

Designing Conversational Agentic Systems

Prompts

Sun, G., Zhan, X., & Such, J. (2024, July). Building better ai agents: A provocation on the utilisation of persona in llm-based conversational agents. In *Proceedings of the 6th ACM Conference on Conversational User Interfaces* (pp. 1-6).

Agent Description

Agents typically have descriptions that include things like their roles, skills/abilities, behavior, etc.

Personas

Planning/Routing/Reasoning

Agentic systems typically have prompts that control the reasoning process, such as how they should decide about what to do. There may also be a prompt that controls whether the system provides thoughts or rationale behind the steps it takes.

Output/ Response

The system may use prompting to determine how the final response is displayed to the user. For example, the system may have used multiple tools to retrieve different pieces of information which then need to be synthesized or compiled into a response.

Tool Description

Tools have descriptions that are used by other components of the system to determine if, and when to use that tool.

Personas

Technical Report

Tencent AI Lab

Scaling Synthetic Data Creation with 1,000,000,000 Personas

Tao Ge,* **Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, Dong Yu**
Tencent AI Lab Seattle
https://github.com/tencent-ailab/persona-hub

Abstract

We propose a novel persona-driven data synthesis methodology that leverages various perspectives within a large language model (LLM) to create diverse synthetic data. To fully exploit this methodology at scale, we introduce Persona Hub – a collection of 1 billion diverse personas automatically curated from web data. These 1 billion personas (~13% of the world's total population), acting as distributed carriers of world knowledge, can tap into almost every perspective encapsulated within the LLM, thereby facilitating the creation of diverse synthetic data at scale for various scenarios. By showcasing Persona Hub's use cases in synthesizing high-quality mathematical and logical reasoning problems, instructions (i.e., user prompts), knowledge-rich texts, game NPCs and tools (functions) at scale, we demonstrate persona-driven data synthesis is versatile, scalable, flexible, and easy to use, potentially driving a paradigm shift in synthetic data creation and applications in practice, which may have a profound impact on LLM research and development.

***DISCLAIMER**: Persona Hub can facilitate synthetic data creation at a billion-scale to simulate diverse inputs (i.e., use cases) from a wide variety of real-world users. If this data is used as input to query a target LLM to obtain its outputs at scale, there is **a high risk** that the LLM's knowledge, intelligence and capabilities will be dumped and easily replicated, thereby challenging the leading position of the most powerful LLMs (e.g., our approach allows a 7B LLM to achieve 65% on MATH, matching the performance of `gpt-4-turbo-preview`). This tech report is **for research purposes only**. It is crucial to avoid misuse and ensure ethical and responsible application. We discuss its broad impact and potential concerns in detail in Section 5.*

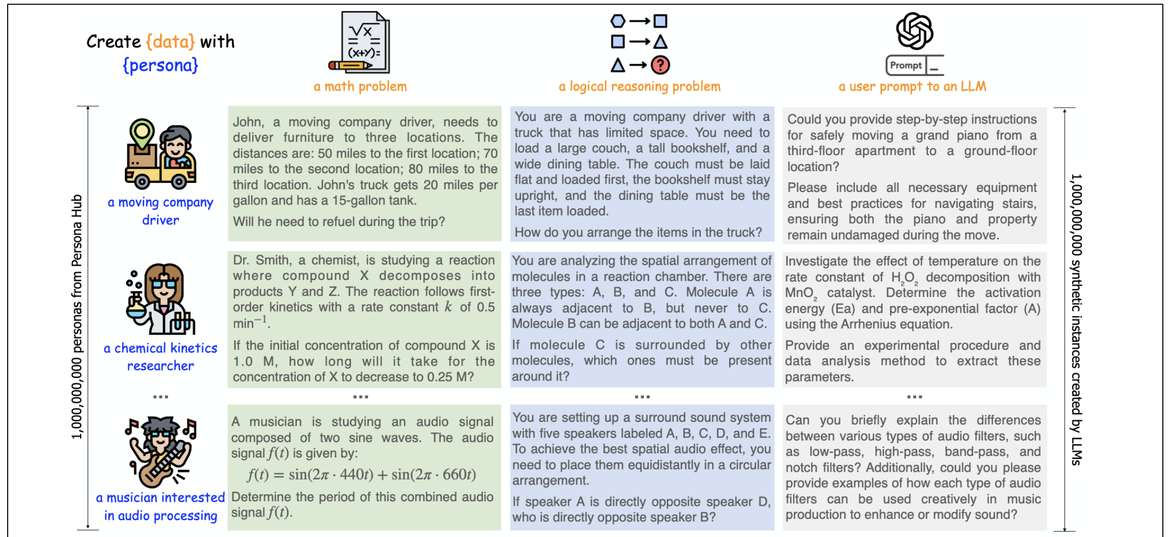


Figure 1: Personas can work with a wide range of data synthesis prompts (e.g., "create a math problem") to guide an LLM to synthesize data with corresponding perspectives. The 1 billion personas in Persona Hub can facilitate various data synthesis scenarios at a billion scale.

Personas

Generative Agent Simulations of 1,000 People

Authors: Joon Sung Park[1]*, Carolyn Q. Zou[1,2], Aaron Shaw[2], Benjamin Mako Hill[3], Carrie Cai[4], Meredith Ringel Morris[5], Robb Willer[6], Percy Liang[1], Michael S. Bernstein[1]

Affiliations:

[1]Computer Science Department, Stanford University; Stanford, CA, 94305, USA.

[2]Department of Communication Studies, Northwestern University; Evanston, IL, 60208, USA.

[3]Department of Communication, University of Washington; Seattle, WA 98195, USA.

[4]Google DeepMind; Mountain View, CA 94043, USA.

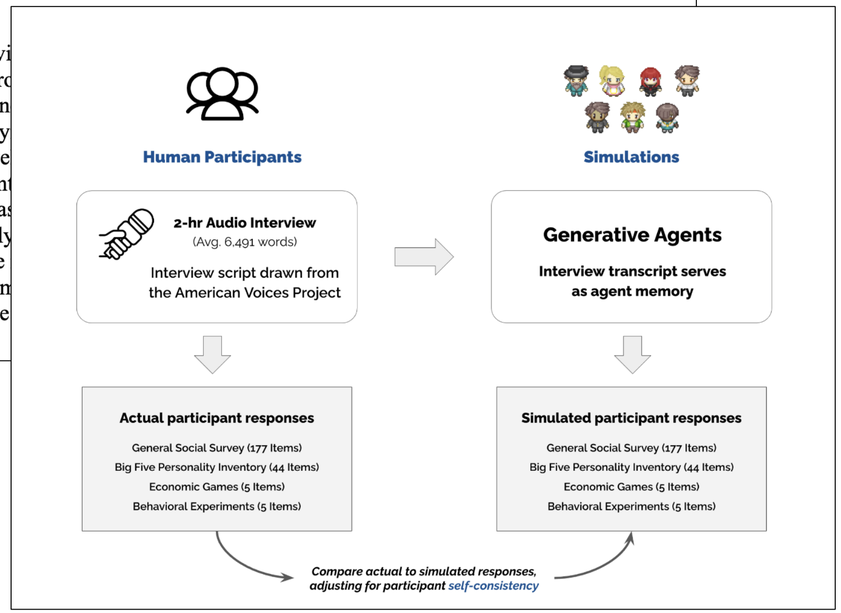
[5]Google DeepMind; Seattle, WA 98195, USA.

[6]Department of Sociology, Stanford University; Stanford, CA, 94305, USA.

*Corresponding author. Email: joonspk@stanford.edu

Abstract:

The promise of human behavi
replicate human behavior acr
social science. We present a n
1,052 real individuals—apply
lives, then measuring how we
individuals that they represent
General Social Survey 85% as
later, and perform comparably
replications. Our architecture
compared to agents given den
tools that can help investigate



Large language models that replace human participants can harmfully misportray and flatten identity groups

Angelina Wang[1], Jamie Morgenstern[2], John P. Dickerson[3,4]

[1]Computer Science, Stanford University, Palo Alto, CA, USA.
[2]Computer Science & Engineering, University of Washington, Seattle, WA, USA.
[3]Computer Science, University of Maryland, College Park, MD, USA.
[4]Arthur, New York City, NY, USA.

Abstract

Large language models (LLMs) are increasing in capability and popularity, propelling their application in new domains—including as replacements for human participants in computational social science, user testing, annotation tasks, and more. In many settings, researchers seek to distribute their surveys to a sample of participants that are representative of the underlying human population of interest. This means in order to be a suitable replacement, LLMs will need to be able to capture the influence of positionality (i.e., relevance of social identities like gender and race). However, we show that there are two inherent limitations in the way current LLMs are trained that prevent this. We argue analytically for why LLMs are likely to both *misportray* and *flatten* the representations of demographic groups, then empirically show this on 4 LLMs through a series of human studies
a third limitation about
each limitation to a perr
that explains why replac
caution in use cases whe
relevant to the task at h
are determined to outwe
engaging human partici
we empirically demonstr

Keywords: large languag
epistemology

Building Better AI Agents: A Provocation on the Utilisation of Persona in LLM-based Conversational Agents

Guangzhi Sun*
University of Cambridge
Cambridge, United Kingdom
gs534@cam.ac.uk

Xiao Zhan*
King's College London
London, United Kingdom
xiao.zhan@kcl.ac.uk

Jose Such
King's College London
London, United Kingdom
& VRAIN, Universitat Politecnica de Valencia, Spain
jose.such@kcl.ac.uk

ABSTRACT

The incorporation of Large Language Models (LLMs) such as the GPT series into diverse sectors including healthcare, education, and finance marks a significant evolution in the field of artificial intelligence (AI). The increasing demand for personalised applications motivated the design of conversational agents (CAs) to possess distinct personas. This paper commences by examining the rationale and implications of imbuing CAs with unique personas, smoothly transitioning into a broader discussion of the personalisation and anthropomorphism of CAs based on LLMs in the LLM era.

We delve into the specific applications where the implementation of a persona is not just beneficial but critical for LLM-based CAs. The paper underscores the necessity of a nuanced approach to persona integration, highlighting the potential challenges and ethical dilemmas that may arise. Attention is directed towards the importance of maintaining persona consistency, establishing robust evaluation mechanisms, and ensuring that the persona attributes are effectively complemented by domain-specific knowledge.

1 WHAT DOES 'PERSONA' MEAN IN THE CONTEXT OF CONVERSATIONAL AGENTS?

In the context of conversational agents (CAs), the concept of *persona* represents the essence or 'soul' of these agents. Persona encapsulates the distinct tone, voice, and personality that characterizes a CA, transforming mechanical interactions into engaging, human-like conversations [18, 46]. Commonly, these attributes of persona can consist any type of information that intend to capture personal characteristics about an individual [28], and are relatively static (race), and slowly change over time (age), or temporary (emotional status) [26, 54].

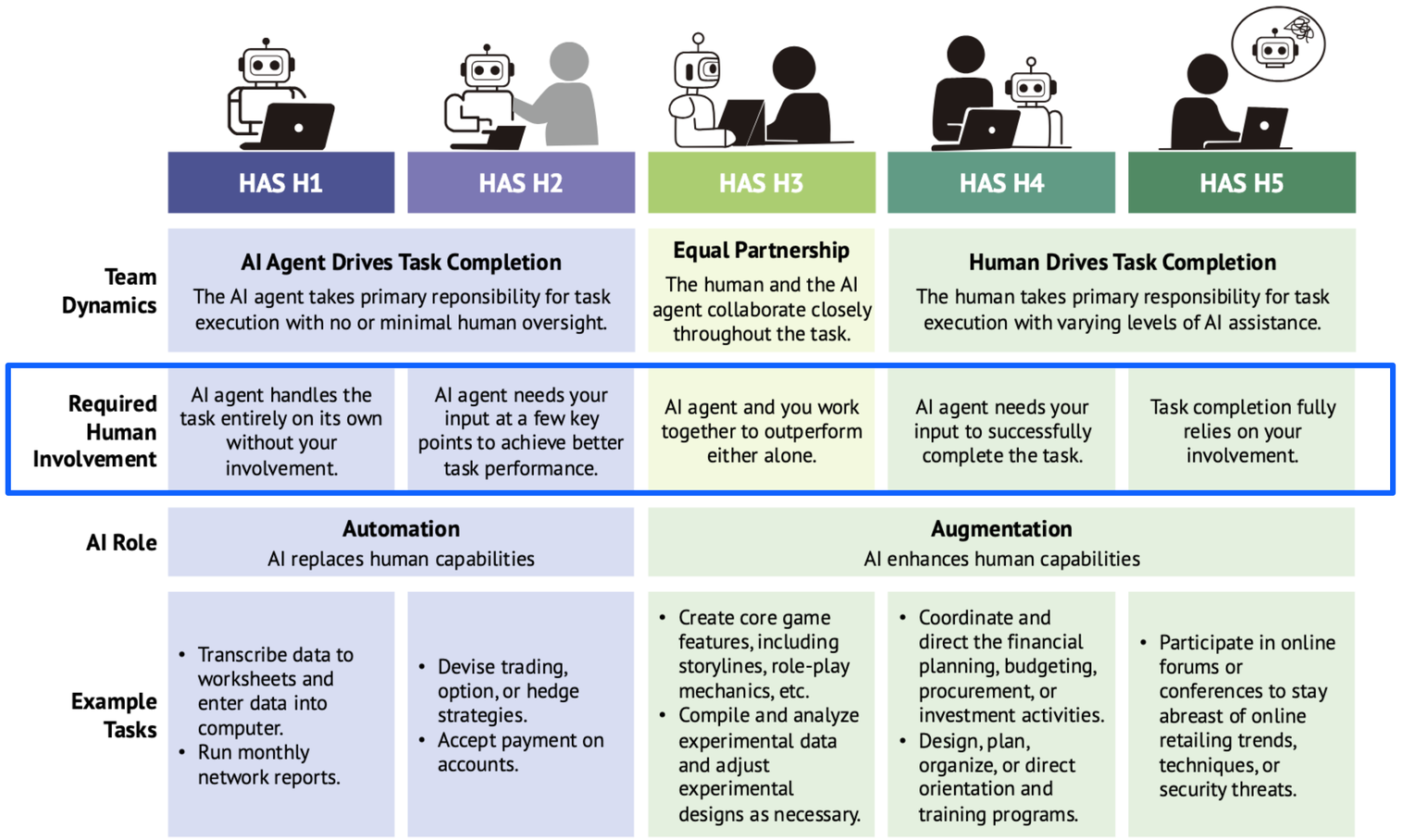
Before delving deeper into the discussion of personas in CAs, it's important to distinguish this concept from the idea of 'personality' that has been explored in prior research [24, 27, 36, 38]. While personality traits, such as being "friendly" or "smart," or frameworks like the Myers-Briggs Type Indicator (MBTI) [4], might define certain characteristics shared by groups of individuals, a persona in CAs represents a more complex and consistent identity [36, 55].

IBM **Research**

Questions for design

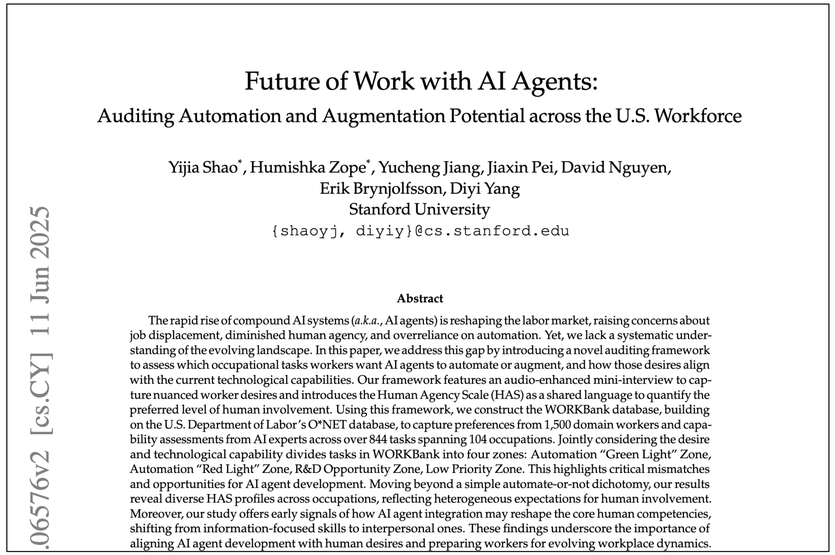
How much information do users want/need about the processing steps?	How much control do users need about how the system behaves?	How fast do users need a response/output?
Agentic systems may have multiple sub-steps to get to an output	Agentic systems can enable users to interact with their reasoning, plan, and actions to intervene or change the plan.	Agentic systems can be slower due to complex reasoning, multiple steps, interaction with external APIs, etc

How much control do users need about how the system behaves?



	HAS H1	HAS H2	HAS H3	HAS H4	HAS H5
Team Dynamics	**AI Agent Drives Task Completion** The AI agent takes primary reponsibility for task execution with no or minimal human oversight.		**Equal Partnership** The human and the AI agent collaborate closely throughout the task.	**Human Drives Task Completion** The human takes primary responsibility for task execution with varying levels of AI assistance.	
Required Human Involvement	AI agent handles the task entirely on its own without your involvement.	AI agent needs your input at a few key points to achieve better task performance.	AI agent and you work together to outperform either alone.	AI agent needs your input to successfully complete the task.	Task completion fully relies on your involvement.
AI Role	**Automation** AI replaces human capabilities		**Augmentation** AI enhances human capabilities		
Example Tasks	• Transcribe data to worksheets and enter data into computer. • Run monthly network reports.	• Devise trading, option, or hedge strategies. • Accept payment on accounts.	• Create core game features, including storylines, role-play mechanics, etc. • Compile and analyze experimental data and adjust experimental designs as necessary.	• Coordinate and direct the financial planning, budgeting, procurement, or investment activities. • Design, plan, organize, or direct orientation and training programs.	• Participate in online forums or conferences to stay abreast of online retailing trends, techniques, or security threats.

Figure 2: **Levels of Human Agency Scale (HAS).** We introduce the Human Agency Scale (*i.e.*, H1-H5) to quantify the team dynamics and degree of human involvement required. HAS provides a shared language to quantify automation vs. augmentation, complementing the traditionally "AI-first" perspective used in defining levels of automation. Importantly, higher HAS levels are not inherently better—different levels suit different AI roles.



Future of Work with AI Agents:

Auditing Automation and Augmentation Potential across the U.S. Workforce

Yijia Shao*, Humishka Zope*, Yucheng Jiang, Jiaxin Pei, David Nguyen,
Erik Brynjolfsson, Diyi Yang
Stanford University
{shaoyj, diyiy}@cs.stanford.edu

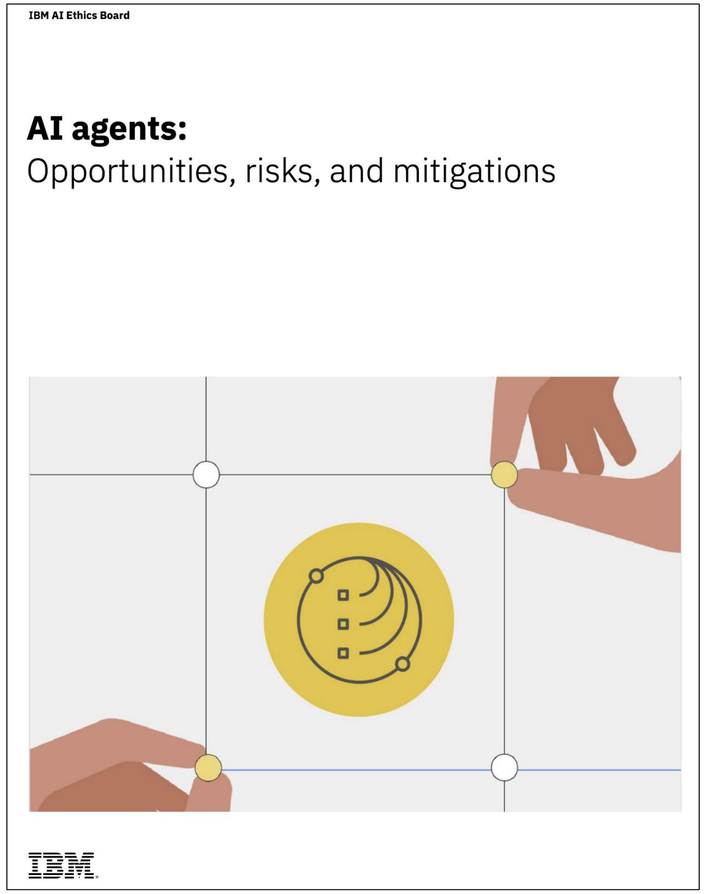
Abstract

The rapid rise of compound AI systems (*a.k.a.*, AI agents) is reshaping the labor market, raising concerns about job displacement, diminished human agency, and overreliance on automation. Yet, we lack a systematic understanding of the evolving landscape. In this paper, we address this gap by introducing a novel auditing framework to assess which occupational tasks workers want AI agents to automate or augment, and how those desires align with the current technological capabilities. Our framework features an audio-enhanced mini-interview to capture nuanced worker desires and introduces the Human Agency Scale (HAS) as a shared language to quantify the preferred level of human involvement. Using this framework, we construct the WORKBank database, building on the U.S. Department of Labor's O*NET database, to capture preferences from 1,500 domain workers and capability assessments from AI experts across over 844 tasks spanning 104 occupations. Jointly considering the desire and technological capability divides tasks in WORKBank into four zones: Automation "Green Light" Zone, Automation "Red Light" Zone, R&D Opportunity Zone, Low Priority Zone. This highlights critical mismatches and opportunities for AI agent development. Moving beyond a simple automate-or-not dichotomy, our results reveal diverse HAS profiles across occupations, reflecting heterogeneous expectations for human involvement. Moreover, our study offers early signals of how AI agent integration may reshape the core human competencies, shifting from information-focused skills to interpersonal ones. These findings underscore the importance of aligning AI agent development with human desires and preparing workers for evolving workplace dynamics.

Shao, Yijia, et al. "Future of Work with AI Agents: Auditing Automation and Augmentation Potential across the US Workforce." *arXiv preprint arXiv:2506.06576* (2025).

Risks and Challenges

- What Can Go Wrong?
- Hallucination and factual errors
- Safety and misuse
- Transparency and trust
- Monitoring and accountability



https://www.ibm.com/granite/docs/resources/ai-agents-opportunities-risks-and-mitigations.pdf

Risks

Group	Risk	Risk Indicator with Reason
Value Alignment	**Misaligned Actions:** AI agents can take actions that are not aligned with relevant human values, ethical considerations, guidelines, and policies. Misaligned actions can occur in different ways such as: • Applying learned goals inappropriately to new or unforeseen situations. • Using AI agents for a purpose/goals that are beyond their intended use. • Selecting resources or tools in a biased way. • Using deceptive tactics to achieve the goal by developing the capacity for • scheming based on the instructions given within a specific context. • Compromising on AI agent values to work with another AI agent or tool to accomplish the task.	Amplified Reason: • Autonomy of AI agents to take actions

Fairness	**Discriminatory actions:** AI agents can take actions where one group of humans is unfairly advantaged over another due to the decisions of the model. This may be caused by AI agents' biases in the actions that impact the world, in the resources consulted, and in the resource selection process. For example, an AI agent can generate code that can be biased.	Amplifed
		
		Reason:
		• Autonomy of AI agents to take actions
		  • Take actions that impact the world
		  • Consulting biased resources
		  • Biased resource selection process

	Data bias: Specific actions taken by the AI agent, such as modifying a dataset or a database, can introduce bias in the resource that gets used by others or by itself to take actions	New Reason: • AI agents taking actions that impact the world. Here, bias is due to a specific action • Open-endedness
Misplaced Trust	**Over- or under-reliance:** Reliance, that is the willingness to accept an AI agent behavior, depends on how much a user trusts that agent and what they are using it for. Over-reliance occurs when a user puts too much trust in an AI agent, accepting an AI agent's behavior even when it is likely undesired. Underreliance is the opposite, where the user doesn't trust the AI agent but should. Increasing autonomy (to take action, select, and consult resources/tools) of AI agents and the possibility of opaqueness and open-endedness increase the variability and visibility of agent behavior leading to difficulty in calibrating trust and possibly contributing to both over- and under-relaince	Amplified Reason: • AI Agents actions make the trust evaluation harder

IBM **Research**

Group	Risk	Risk Indicator with Reason
Computation Inefficiency	**Redundant actions:** AI agents can execute actions that are not needed for achieving the goal. These actions could result in wasting computation resources, reducing AI agent's efficiency in achieving the goal, and leading to potentially harmful outcomes. In an extreme case, AI agents could enter a cycle of executing the same actions repeatedly without any progress. This could happen because of unexpected conditions in the environment, the AI agent's failure to reflect on its action, AI agent reasoning and planning errors or the AI agent's lack of knowledge about the problem. This could prevent the AI agent from achieving the goal and exhaust computational resources.	New Reason: • AI agents can take actions
Robustness	**Attack on AI agents' external resources:** Attackers intentionally create vulnerabilities or exploit existing vulnerabilities in external resources (tools/database/applications/services/other agents) that AI agents rely on to execute their intended actions or to achieve their goals. Compromised external resources could impact the AI agent's performance in different ways, such as: • Manipulating AI agents to pursue a different goal. Example: Change AI agents' goal to add positive reviews to a product of attacker's preference when the original user goal is to add queries about the product. • Manipulating AI agents to execute undesired actions. Example: Trick AI agents to download malware. • Capturing and relaying interactions between AI agents to malicious actors. • Getting AI agents to share personal or confidential information.	New Reason: • AI agents can take actions • Open-endedness and complexities due to open-endedness • Access to more resources
	Unauthorized use: If attackers can gain access to the AI agent and its components, they can perform actions that can have different levels of harm depending on the agent's capabilities and information it has access to. Additionally, attackers may execute actions that lead to system degradation, such as exhausting available resources and impairing performance. Examples: • Using stored personal information to mimic identity or impersonate with an intent to deceive. • Manipulating AI agent's behavior via feedback to the AI agent or corrupting its memory to change its behavior. • Manipulating the problem description or the goal to get the AI agent to behave badly or run harmful commands.	Amplified Reason: • AI agents can take actions • AI agents are more capable • Personalization feature of AI agents
	Exploit trust mismatch: Attackers could initiate injection attacks to bypass the trust boundary, which is a distinct point or conceptual line where the level of trust in a system, application, or network changes. This could lead to mismatched (expected vs. realized) trust boundaries and could result in unintended tool use, excessive agency, and privilege escalation. Also, background execution in multi-agent environments increases the risk of covert channels if input/output validation is weak.	Amplified Reason: • Open-endedness • Complexity
	Function-calling hallucination: AI agents could make mistakes when generating function calls (calls to tools to execute actions). Those function calls could result in incorrect, unnecessary or harmful actions. Examples: Generating wrong functions or wrong parameters for the functions.	New Reason: • AI agents can consult resources and tools • AI agents can take actions

Group	Risk	Risk Indicator with Reason
Privacy and IP	**Sharing IP/PI/confidential information with user:** AI agents with unrestricted access to resources or databases or tools could potentially store and share PI/IP/confidential information with system users when performing their actions.	Amplified Reason: • Multi-component nature with the ability to perform actions
	Sharing IP/PI/confidential information with tools: AI agents with unrestricted access to resources or databases or tools could potentially store and share PI/IP/confidential information with other tools or agents when performing their actions.	New Reason: • Multi-component nature with the ability to perform actions
Explainability and Transparency	**Unexplainable and untraceable actions:** Explanations, lineage and trace information, and source attribution for AI agent actions may be difficult, imprecise, or unobtainable.	Amplified Reason: • Difficult to trace cause and effect or influence of different components including LLMs across final actions
	Lack of transparency: Lack of transparency is due to insufficient documentation of the AI agent design, development, evaluation process, absence of insights into inner workings of the AI agent, and interaction with other agents/tools/resources.	Amplified Reason: • Rely on other documents available for other tools/agents

https://www.ibm.com/granite/docs/resources/ai-agents-opportunities-risks-and-mitigations.pdf

IBM **Research**

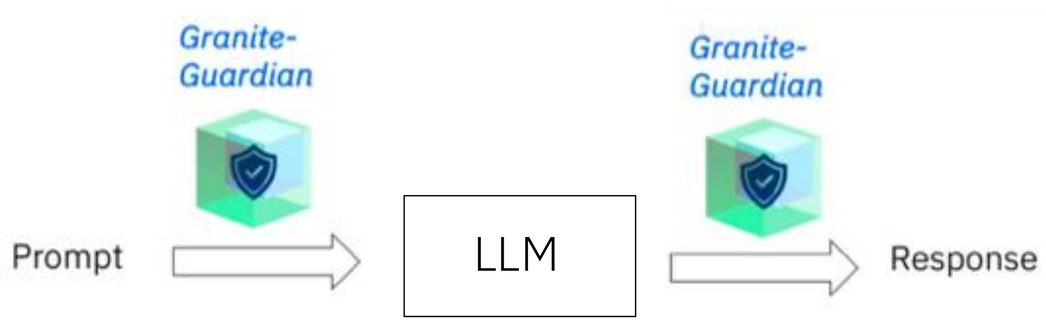
Challenges

Challenge	Challenge Indicator with Reason
Evaluation: Challenge in evaluating AI agents' performance/ accuracy because of the complexity of the system and open-endedness.	Amplified Reason: • Complexity of the agentic systems
Mitigation and maintenance: Challenge in knowing where in the system something is going wrong and how to fix it or what might need maintenance protocols.	Amplified Reason: • Complexity and open-endedness of the agentic systems
Reproducibility: Challenge in reproducing the agent's behavior or output because of unavailability or changes to the tools or resources used for executing the actions.	New Reason: • Open-endedness
Accountability: Challenge in assigning responsibility for an action taken by an agentic AI system.	Amplified Reason: • Complexity & open-endedness. Components can be from different vendors
Compliance: Challenge in determining regulatory compliance as AI agents are complex and there may not be enough information to understand whether the whole agentic AI system is compliant.	Amplified Reason: • Open-endedness • Complexity • Lack of transparency

https://www.ibm.com/granite/docs/resources/ai-agents-opportunities-risks-and-mitigations.pdf

Granite guardian

The Granite Guardian models are a collection
of models designed to detect risks in prompts and responses.



User input risks

- **Social Bias:** Prejudice based on identity or social characteristics
- **Jailbreaking:** Manipulation of AI to generate harmful, unwanted, or inappropriate content
- **Violence:** Content that promotes physical, mental, or sexual harm
- **Profanity**: Use of offensive language or insults
- **Unethical Behavior**: Violations of moral or legal standards
- **Personally Identifiable Information:** Sensitive data that can be used to identify an individual

Granite Guardian

Inkit Padhi[†] Manish Nagireddy[*] Giandomenico Cornacchia[*]
Subhajit Chaudhury[*] Tejaswini Pedapati[*] Pierre Dognin
Keerthiram Murugesan Erik Miehling Martin Santillan Cooper
Kieran Fraser Giulio Zizzo Muhammad Zaid Hameed Mark Purcell
Michael Desmond Qian Pan Zahra Ashktorab Inge Vejsbjerg
Elizabeth Daly Michael Hind Werner Geyer
Ambrish Rawat[†] Kush R. Varshney[†] Prasanna Sattigeri[†]

IBM Research
inkpad@ibm.com, ambrish.rawat@ie.ibm.com, krvarshn@us.ibm.com,
psattig@us.ibm.com

2024

Padhi, Inkit, et al. "Granite guardian." *arXiv preprint arXiv:2412.07724* (2024).

LLM output risks

- **Input Risks:** Same risks associated with input (Social Bias, Violence, Profanity, and Unethical Behavior), with the exception* of Jailbreaking
- **Context Relevance:** A hallucination where the retrieved context is not relevant to answering the user's question or meeting their needs.
- **Groundness:** A hallucination where the model's response includes claims not supported by the provided context.
- Answer Relevance: The response does not adequately address or respond to the user's input.
- Function Call Hallucination: The assistant's response contains function calls that have syntax or semantic errors.

https://www.ibm.com/architectures/product-guides/granite-guardian

IBM **Research**

Frameworks for Agentic Systems

Agentic frameworks/ platforms

LangGraph

“LangGraph provides low-level supporting infrastructure that sits underneath any workflow or agent.”

IBM BeeAI

“The open-source platform to discover, run, and compose AI agents from any framework”

Microsoft Autogen

“A framework for building AI agents and applications”

CrewAI

“Unlock the Power of Multi-Agent AI With CrewAI - The Leading Open SourcePlatform”

LlamaIndex

“LlamaIndex is the leading framework for building LLM-powered agents over your data with LLMs and workflows.”

OpenAI Agents SDK

“The OpenAI Agents SDK is a lightweight yet powerful framework for building multi-agent workflows.”

Technical considerations for choosing a framework/ architecture

When, if at all, should the system use agentic capabilities?

Agentic processing can be slow and costly. Does your system require an agentic workflow and if so, should it use the agentic workflow all the time?

Context/Memory

The amount of context and/or memory needed and how it functions may impact the kind of framework/architecture needed, as some have more or less support

Single vs. multi-agent

- Single agents are simpler, more predictable, faster, and cheaper (best for simple chatbots and recommendation systems)
- Multi-agent systems are more flexible and have specialized agents but are more complex, expensive, and slower

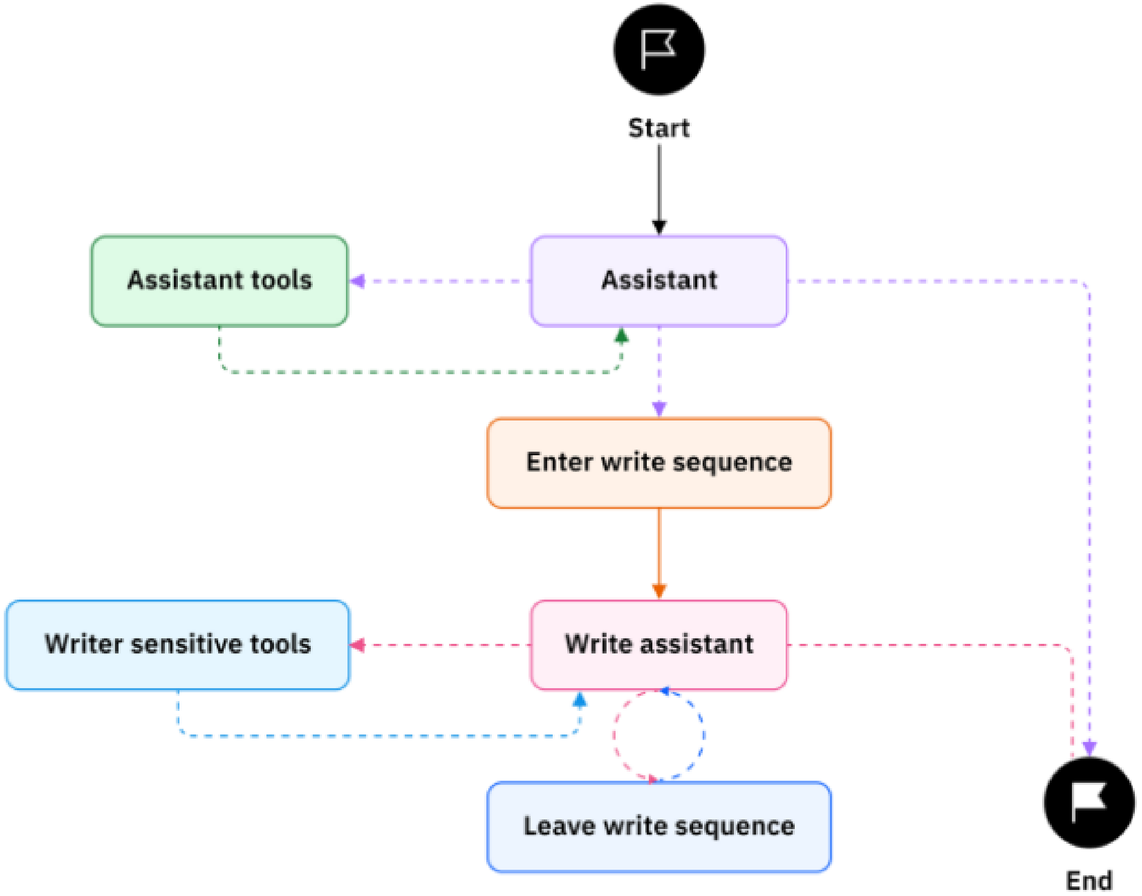
Models

Which model or models you plan to use impacts which frameworks you can work with.

How to build one
Agent Frameworks

IBM **Research**

LangGraph



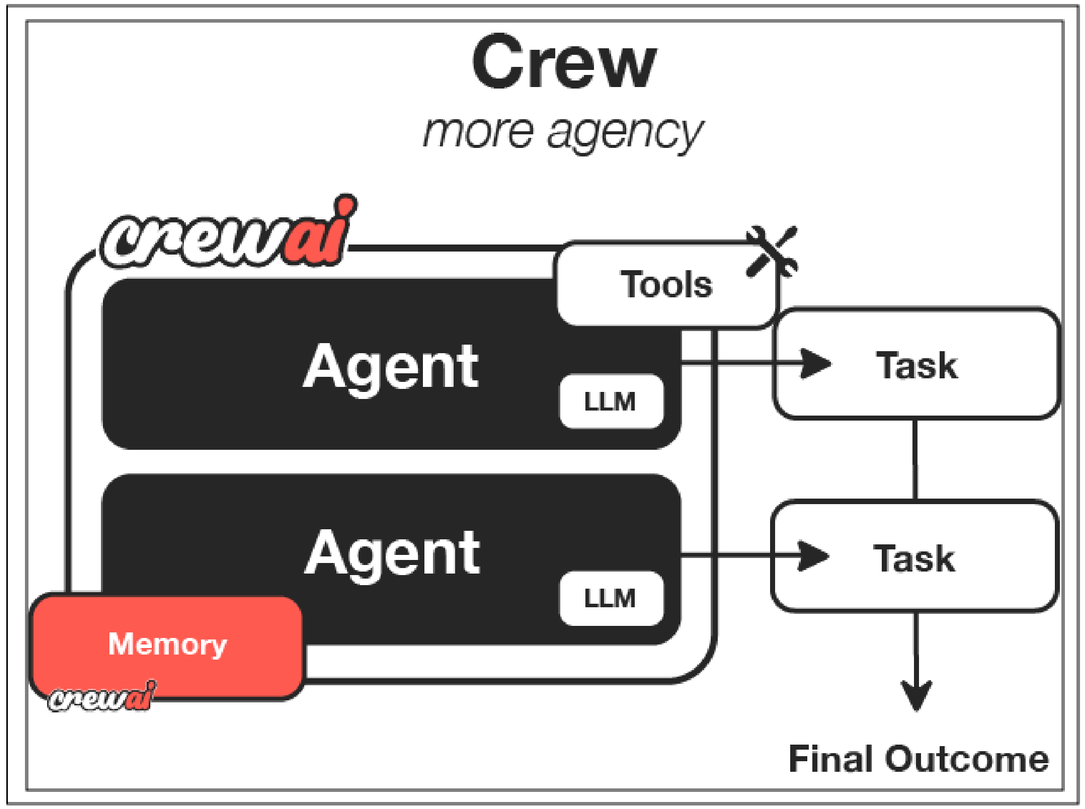
https://www.ibm.com/think/topics/langgraph

LanGraph lets you create graph-based workflows, so you can model agentic systems as a series of clear, making agent nodes and decision nodes.

- Problem: Hard to orchestrate stateful, multi-agent flows
- Benefit: Graph nodes & edges define roles, data, and events. Precise control, easy visualization, and fault-tolerant execution
- Use Case: Perfect for multi-step tasks in production

IBM **Research**

Reference for comparison of frameworks: link

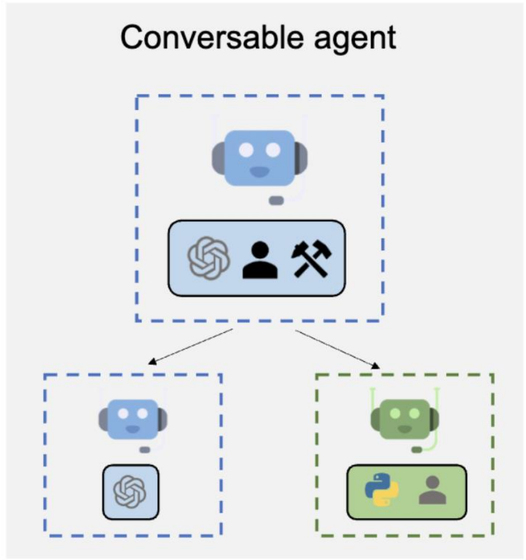


Crew AI models each agent's responsibility as a role and fires events to drive the workflow—ensuring delegation.

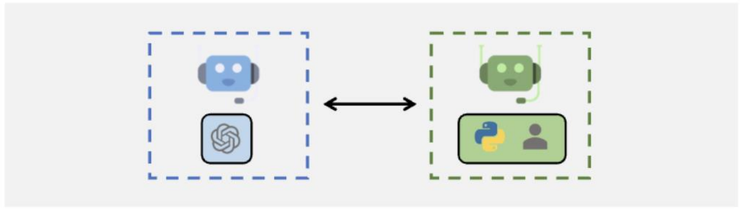
- Problem: Complex multi-agent flows are hard to coordinate
- Benefit: Role assignments + event triggers for deterministic handoffs, orchestration, single-call LLM precision, fast prototyping
- Note: Requires developer expertise for setup

Reference for comparison of frameworks: link

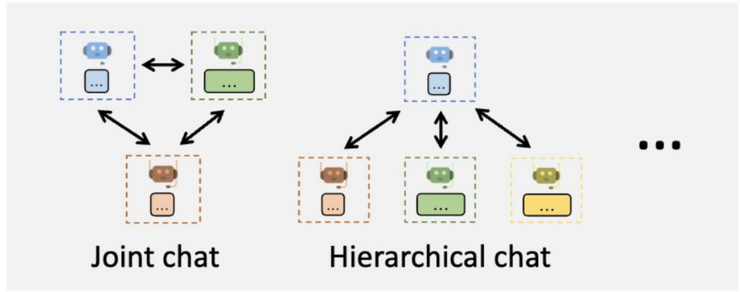
AG



Agent Customization



Multi-Agent Conversations



Flexible Conversation Patterns

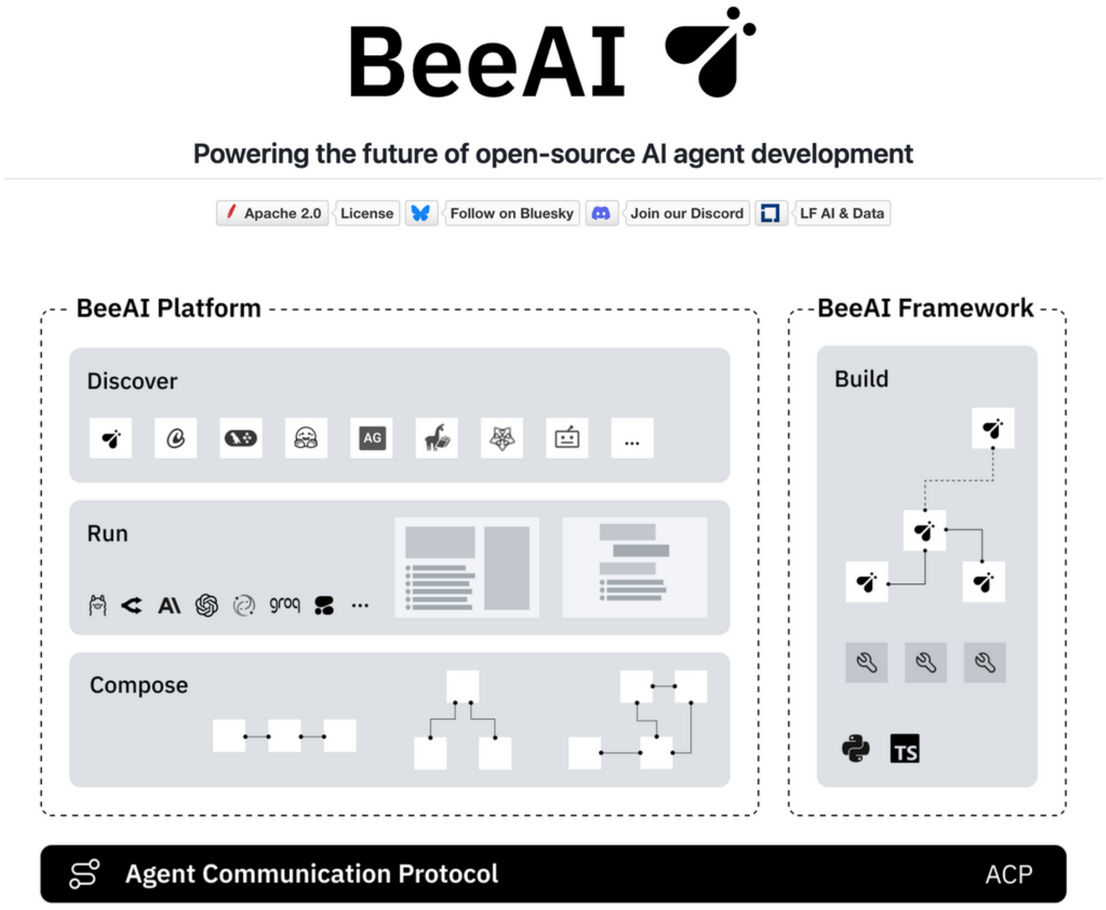
AutoGen lets you orchestrate complex conversations with precision, developers get full control over chat logic and can integrate new capabilities, all backed by strong error resilience.

- Problem: Hard to maintain complex, multi-turn chatbot workflows
- Benefit: Customizable agents and nested chats via simple prompts or code, Dynamic adaptation with tools (e.g., code exec, web access) and human input
- Outcome: Reliable, enterprise-ready conversational systems

https://jrodthoughts.medium.com/inside-autogen-microsoft-research-new-autonomous-agent-framework-b413648af24d

IBM **Research**

Reference for comparison of frameworks: link



https://github.com/i-am-bee

BeeAI is an open-source platform designed to simplify the discovery, execution, and sharing of AI agents—regardless of the framework they're built on.

BeeAI provides a standardized environment to bring your agents to life and collaborate seamlessly.

With BeeAI, you can build agentic systems using any of the frameworks mentioned earlier—and then bring them all together in one cohesive ecosystem.

Framework selection

To explore how agents and humans interact, collaborate, and fluidly switch roles, AutoGen stands out as the ideal framework.

It is purpose-built to support mixed-initiative workflows, where both humans and AI agents can take the lead, respond, and adapt dynamically.

Gives more Human agency.

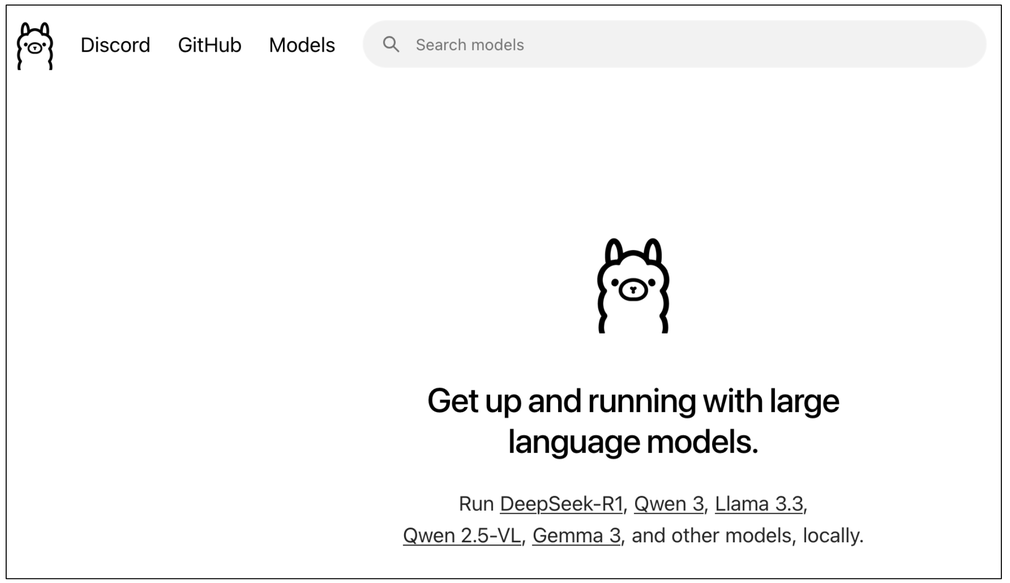
Model access

- Where can I find and access the models?
- Locally (e.g. Ollama, HuggingFace) or API call (e.g. OpenAI service; Watson.ai)
- Should I care about data privacy?
- Should I pay for that?

Ollama - Local LLM

Autogen allows you to connect with services like open-ai and other llm providers. **Here we chose an open-sourced tool called Ollama**

Ollama is a tool that allow you to have llms on your own private local device.



https://ollama.com/blog/ibm-granite

The agent brain
Choosing the right LLM

deepseek-r1
DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.
tools thinking 1.5b 7b 8b 14b 32b 70b 671b
50.4M Pulls 35 Tags Updated 1 week ago

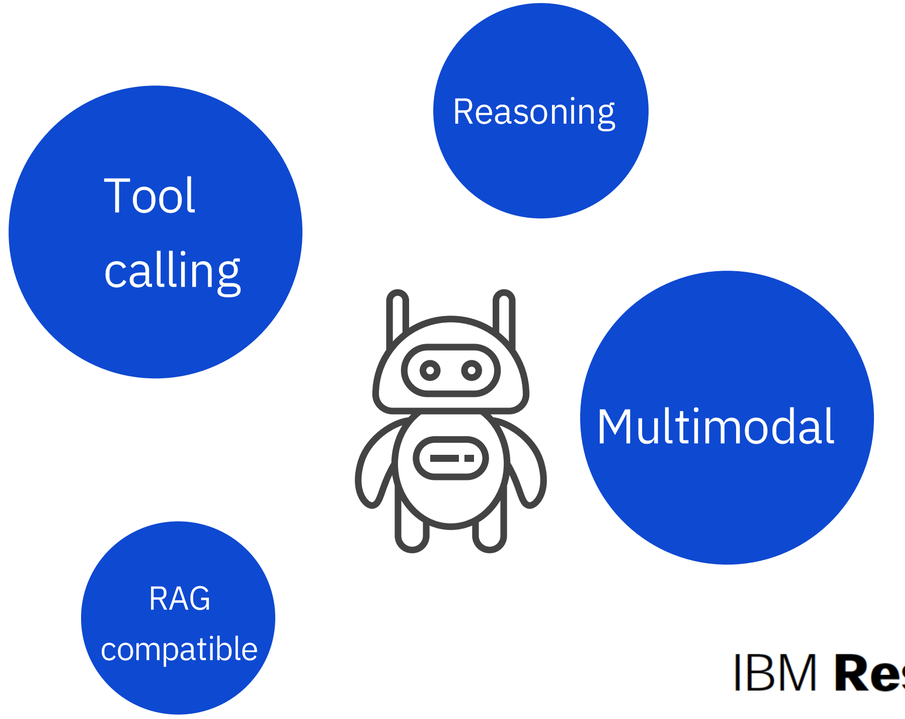
qwen3
Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.
tools thinking 0.6b 1.7b 4b 8b 14b 30b 32b 235b
2.9M Pulls 35 Tags Updated 3 weeks ago

Meta

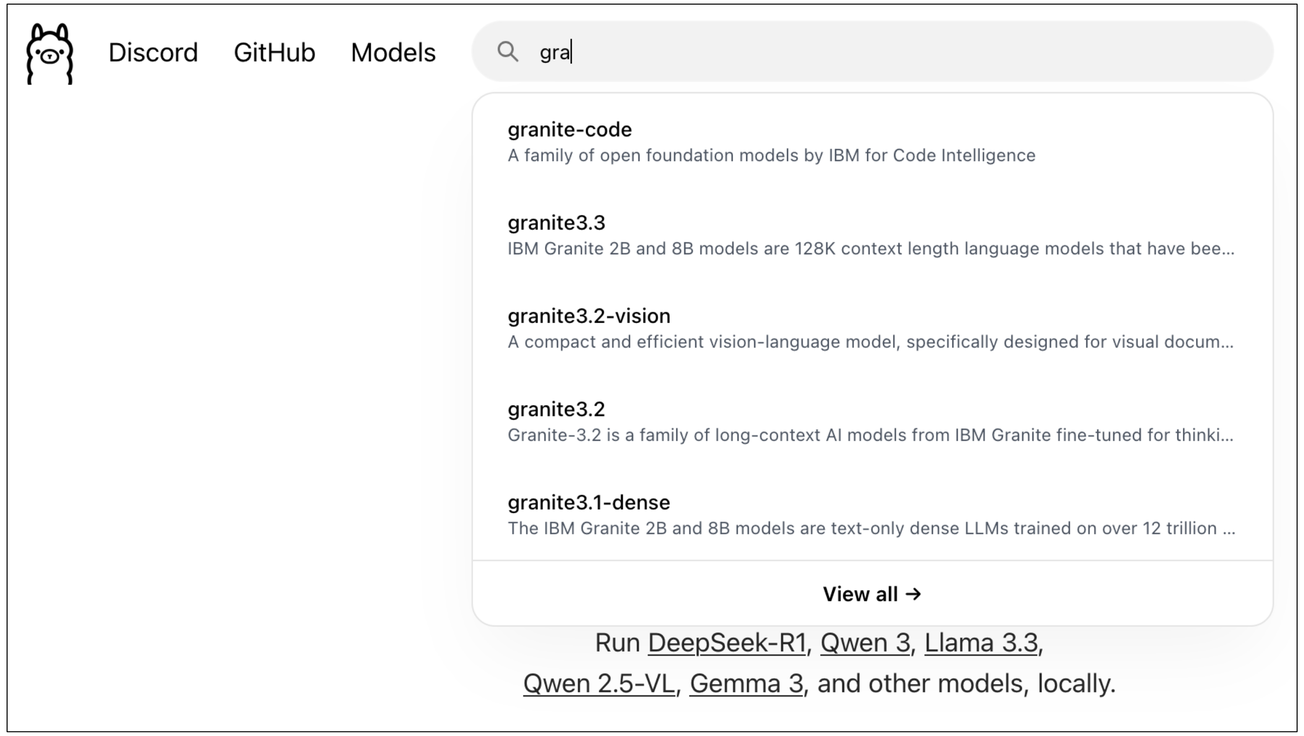
llama4
Meta's latest collection of multimodal models.
vision tools 16x17b 128x17b
434.7K Pulls 11 Tags Updated 1 week ago

Ollama gives you access to many LLMs

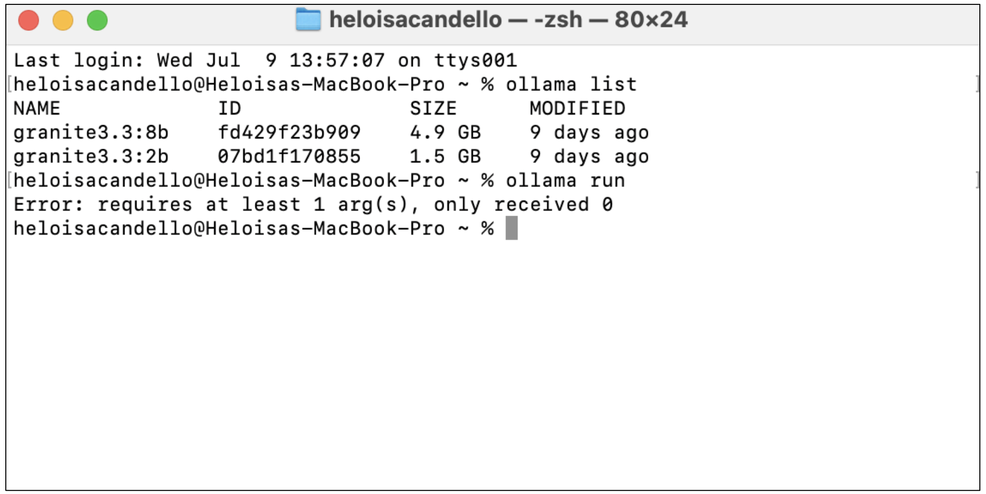
You are choosing the “brain” of the agents in your system. Ensure it handles complex reasoning, retrieval, and API calls.

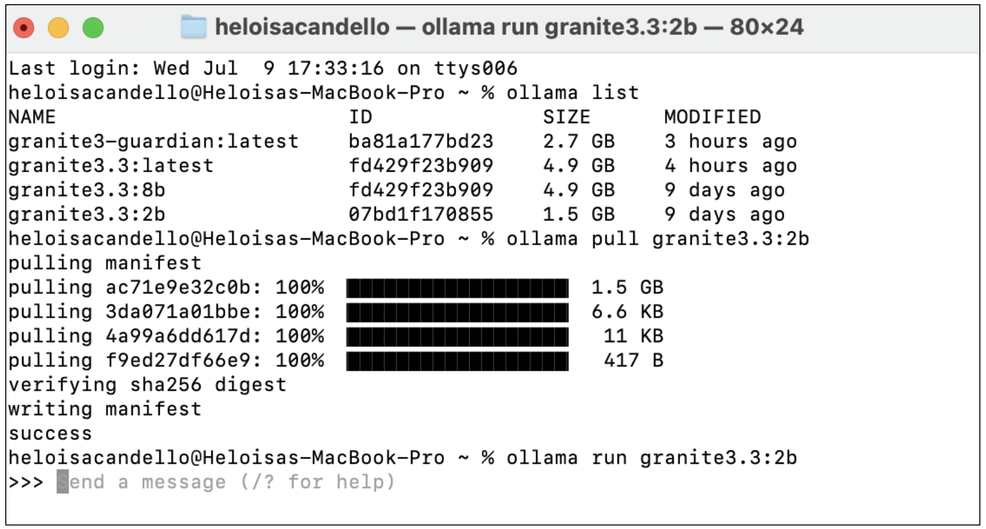


IBM **Research**



https://ollama.com/blog/ibm-granite





Terminal

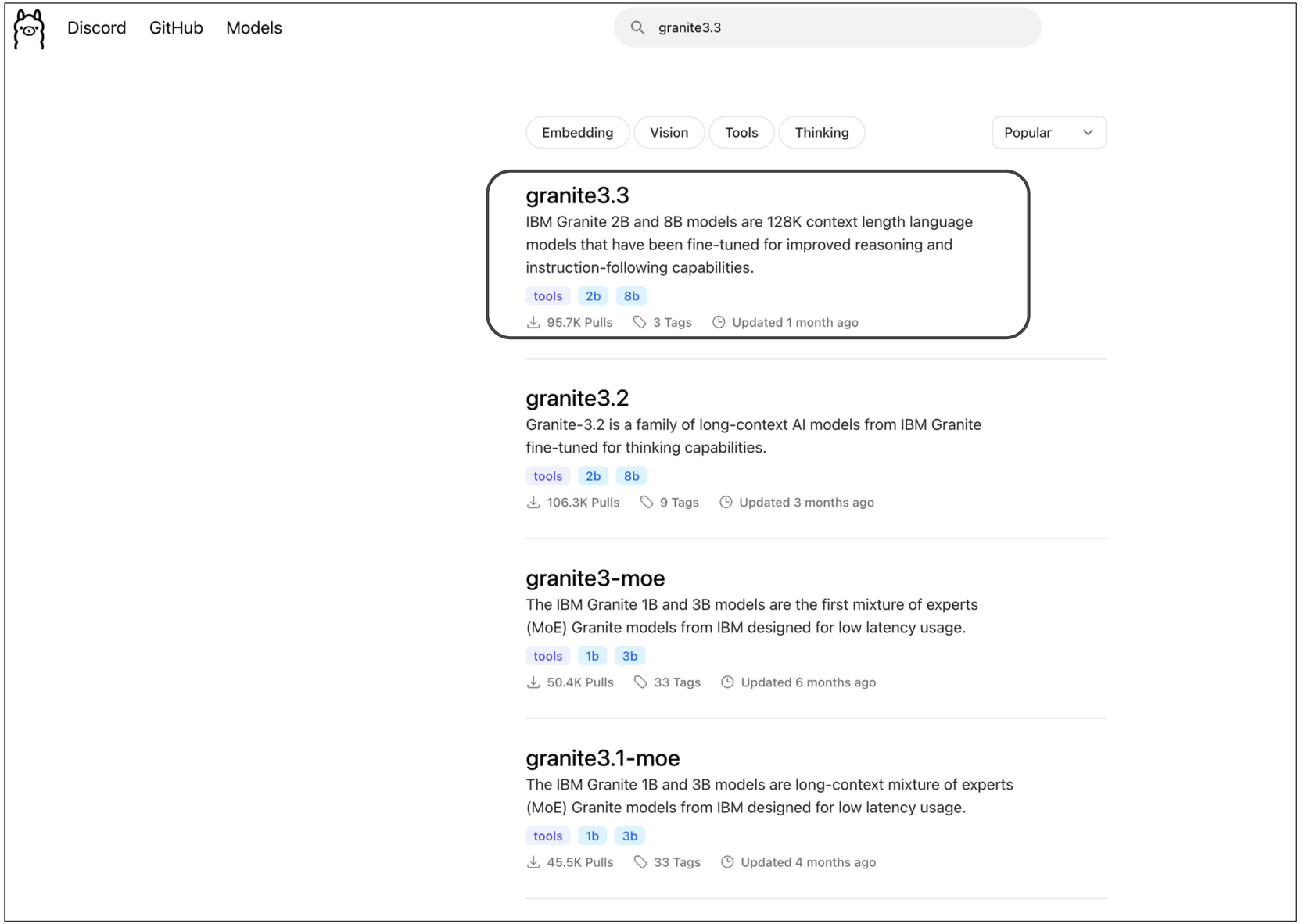
IBM **Research**

IBM’s Granite models

.

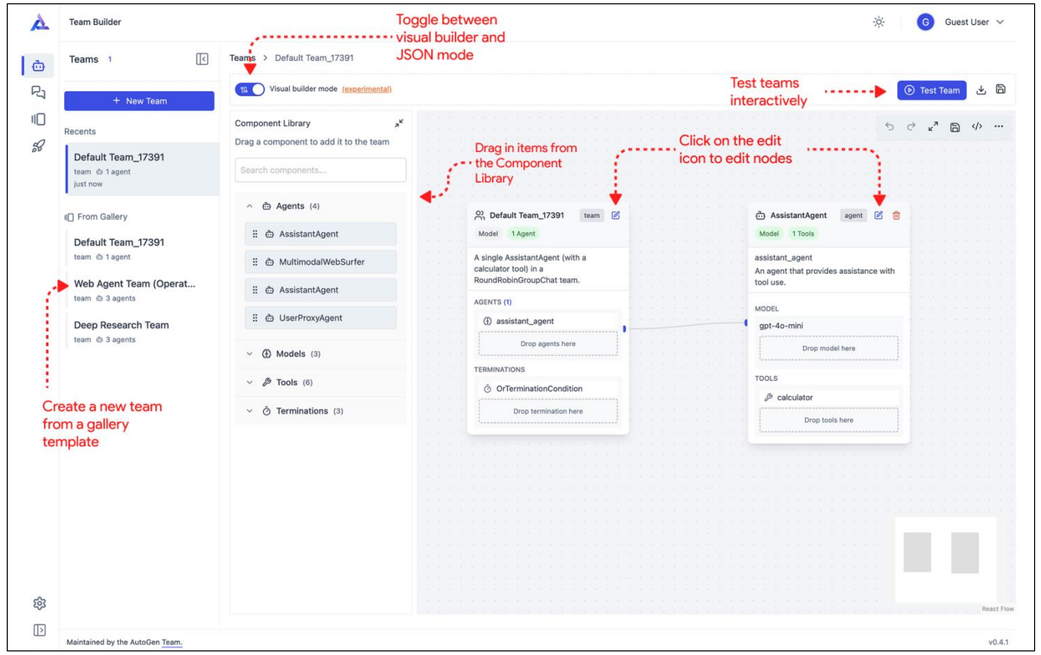
Granite 3.3 - agent capabilities: reasoning, retrieval, tool-calling — without taxing your hardware!

You can look into the details in the [ollama page]



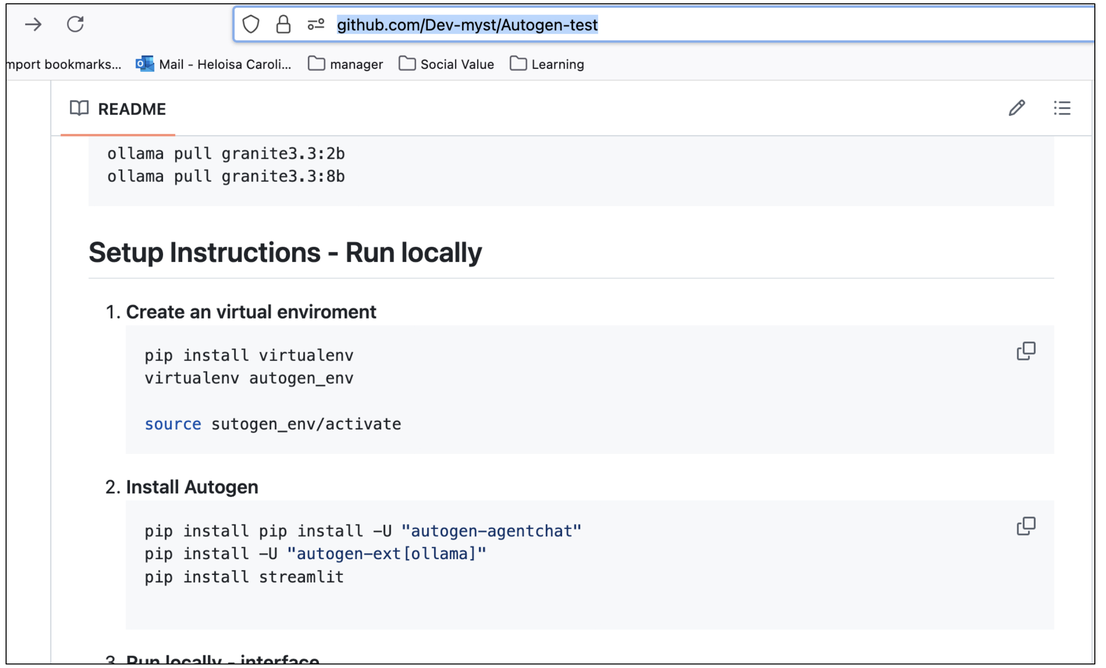
https://ollama.com/search?q=granite3.3

Autogen



Autogen studio

https://microsoft.github.io/autogen/0.2/docs/Getting-Started/



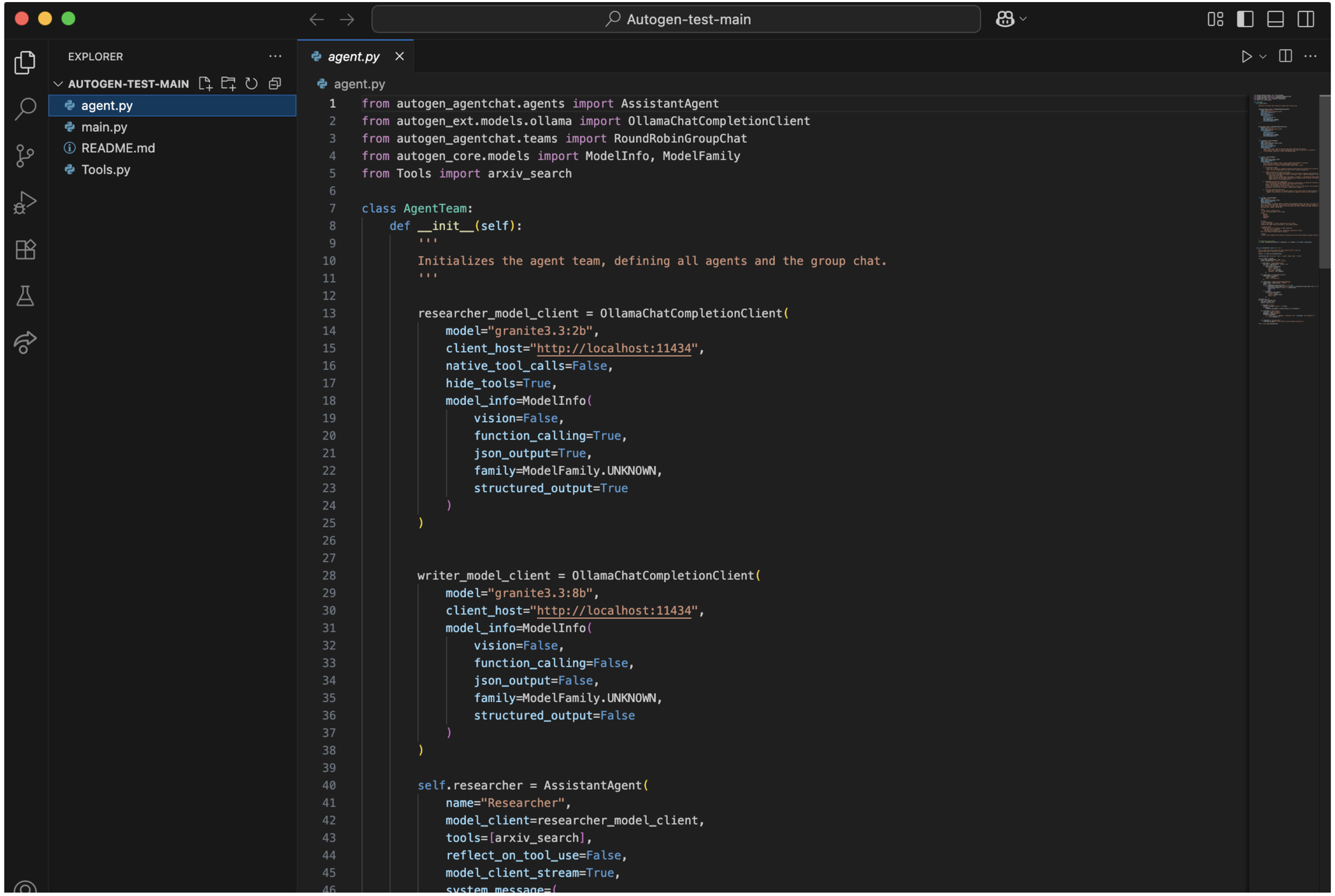
Autogen programming framework

Our tutorial: https://github.com/Dev-myst/Autogen-test/tree/main

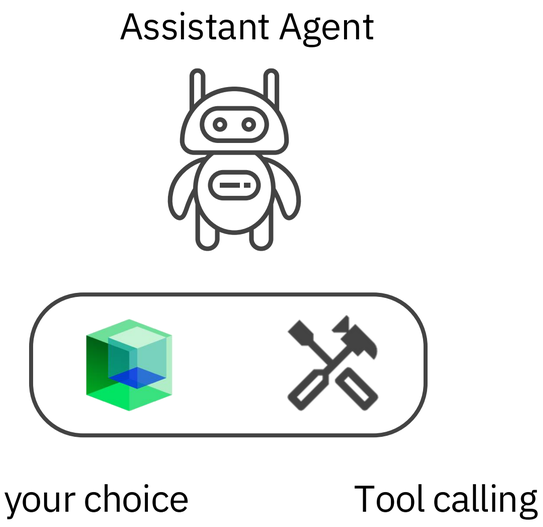
IBM **Research**

Visual Studio Code

Integrated development environment



AgentChat
Main agent structure

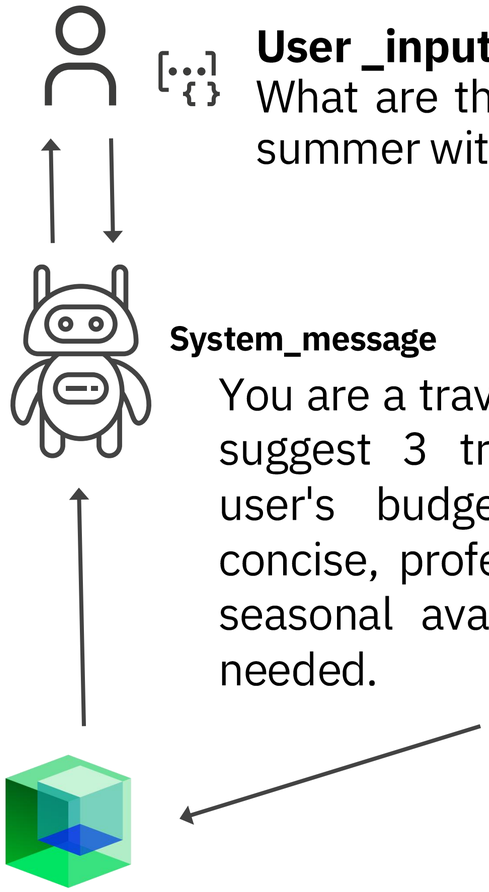


The AssistantAgent is the main agent type in AutoGen. It’s designed for educational and prototyping purposes, helping you explore the framework without complexity.

Giving context to our agent: Prompt engineering

We shape agent behavior both from inside (via system_message) and outside (via user input).

The system defines how it acts; the user defines what it acts on.



User _input:
What are the best places to travel in the summer with low-budget?

System_message

You are a travel planner assistant. Your job is to suggest 3 travel destinations based on the user's budget and preferred activities. Be concise, professional, and prioritize safety and seasonal availability. Ask for more details if needed.

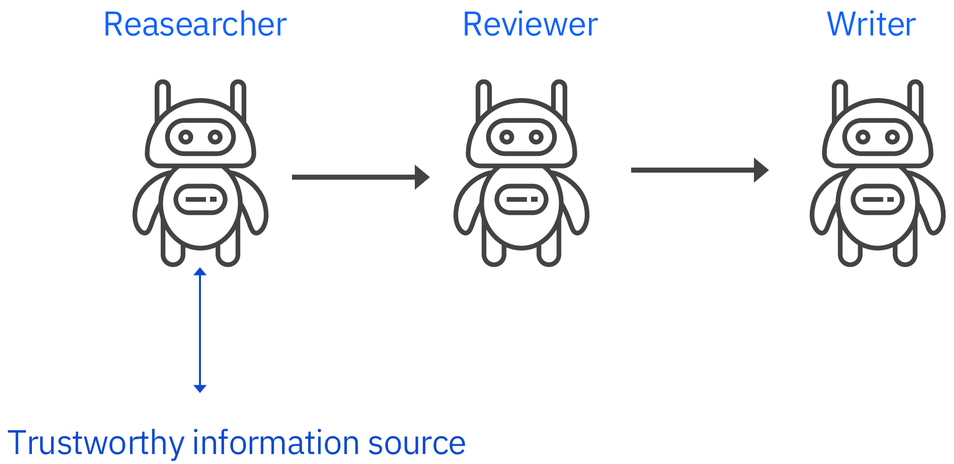
task (user input) + system_message

We are telling our agent:

- a role: travel planner
- behavior: concise, professional, safety-focused
- flexibility: ask details

Important to note that this is inside our agent engines, we do it when coding our agent.

Demo:
Literature review workflow



Let’s build an agentic system that automates the literature review process.

The goal is to search for academic papers from reliable sources based on a user query and turn those into a structured summary.

Our system will consist of three agents:

- **Researcher:** searches for papers using trustworthy sources.
- **Reviewer:** verifies the quality and relevance of the results.
- **Writer:** summarizes the literature search.

Agents
Model configuration

```
researcher_model_client = OllamaChatCompletionClient(
    model="granite3.3:2b",
    client_host="http://localhost:11434",
    native_tool_calls=False,
    hide_tools=True,
    model_info=ModelInfo(
        vision=False,
        function_calling=True,
        json_output=True,
        family=ModelFamily.UNKNOWN,
        structured_output=True
    )
)
```

The Granite used for the Researcher will be the granite 3.3:2b.

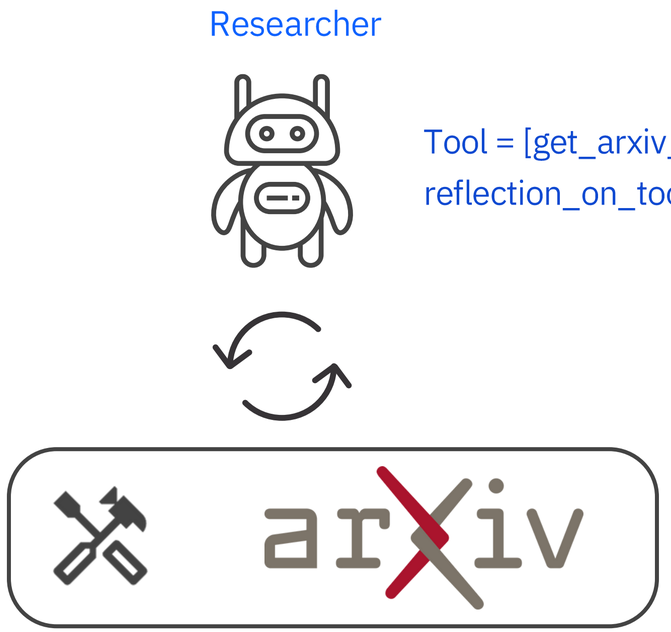
We use a small model because this agent will mainly deal with tool calling.

We disabled native tool calling (native_tool_calls=False) from Granite to make sure that only our own custom tools are used (manual tool calling).

This removes ambiguity and gives us full control over the tool interaction logic.

- More reliable
- More predictable behavior
- Only our custom tools, nothing hidden

Researcher agent tool configuration



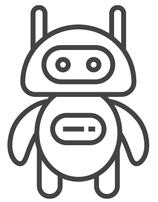
We set up a **search tool for our agent to look for papers**, the **ArXiv API**, which allows for an easy and simple configuration.

Another important point is deciding what we want our agent to do if the tool doesn't work.

Should it try again? Or just stop?

By setting reflection_on_tool_use = True, we allow the agent to analyze the result of the tool call. If the call fails or the results aren't good enough, the agent might try again.

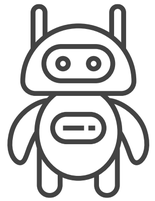
Researcher agent - tool and prompt configuration




```
)

self.researcher = AssistantAgent(
    name="Researcher",
    model_client=researcher_model_client,
    tools=[arxiv_search],
    reflect_on_tool_use=False,
    model_client_stream=True,
    system_message=(
        'given a user topic or question, search of relevant arXiv query. When the tool returns'
        'choose exactly the number of papers requested and pass them as JSON with all information '
        'to the Reviewer, make sure to return the PDF_URL field'
    )
)
```

Reviewer agent - configuration

Reviewer

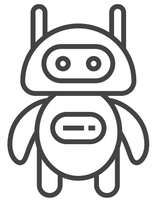


```
self.reviewer = AssistantAgent(
    name="Reviewer",
    model_client=writer_model_client,
    model_client_stream=True,
    system_message='''
    You are a reviewer of academic papers retrieved by the Researcher Agent. Your goal is to check if the selected papers are aligned with the
    user's query and if they cover the key points of interest.Keep your response clear and focused, using only the information available in the abstracts.
    Select the most relevant papers based on the user's query and return exactly the number requested. Each paper must be formatted with its title, authors,
    published date, summary, and PDF URL.

    Given:
    - A user topic or research query.
    - A list of selected papers in the format:
    Title: ,
        PDF URL: ,
        Authors: ,
        Published: ,
        Summary: ,


    You must:
    1. Assess Relevance:
    -Check if each paper is clearly connected to the user query.
    -Identify each paper that are off-topic or only loosely related.

    2. Coverage Check:
    - Briefly state if the collection of papers addresses:
        - The main topic or question.
        - Sub-topics or dimensions (e.g., methodology, application, theory).
    Note if any important angles might be missing.

    3.Output:
    - Write a short paragraph (3–5 sentences) evaluating the overall match between the papers and the user query and attention key points.

    '''
)
```

No

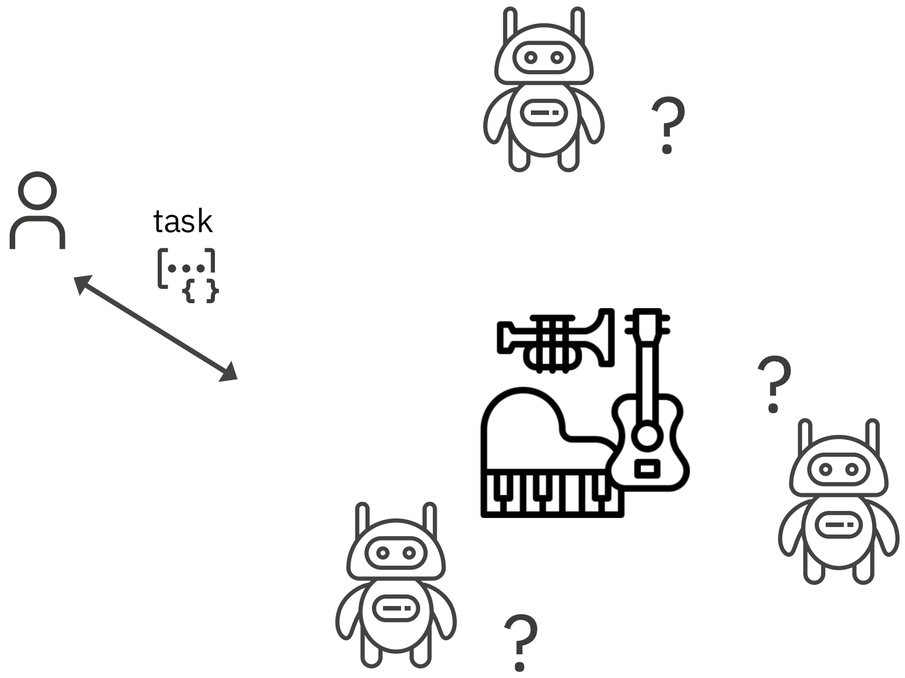
Writer agent - configuration

Writer



```
self.writer = AssistantAgent(
    name="Writer",
    model_client=writer_model_client,
    model_client_stream=True,
    system_message='''
        You are an expert academic writer, write a cohesive and insightful text based exclusively on the provided papers information.
        You will follow these steps in order:

        1.  Introduction & Scope:
            - Start with a concise 2-3 sentence introduction that defines the research area covered by the papers.
            - Then, list the reviewed papers with their titles, authors and paper_url.

        2.  Thematic Synthesis (The Core of Your Task):
            - Identify 2-3 central themes that emerge from the collection of abstracts make references of the papers you are taking information from.
            - For each theme, including the reference of the paper, create a separate paragraph. In each paragraph:
                - Clearly state the theme.
                - Detail how each relevant paper contributes to, defines, or challenges the theme asked for the user.
                - Compare and contrast the papers approaches and findings within the theme. Use phrases like "While Paper A focuses on...,
                  Paper B offers a contrasting view by...".

        3.  Methodological Overview & Limitations :
            - Briefly summarize the methodologies mentioned in the abstracts, including the reference to the papers
            (e.g., "The reviewed studies employ a mix of qualitative analysis,
            machine learning models, and user surveys...").
            - Point out any potential limitations or gaps that can be inferred *solely from the abstracts* (e.g., "A potential
            gap appears to be the lack of focus on long-term user studies...").

        4.  Conclusion and Future Directions:
            - Conclude with a powerful 1-2 sentence summary of the current state of research based on these papers.
            - Suggest a key direction for future research that logically follows from your analysis.

            '''
```

Building Agentic System: Orchestra or Chaos?



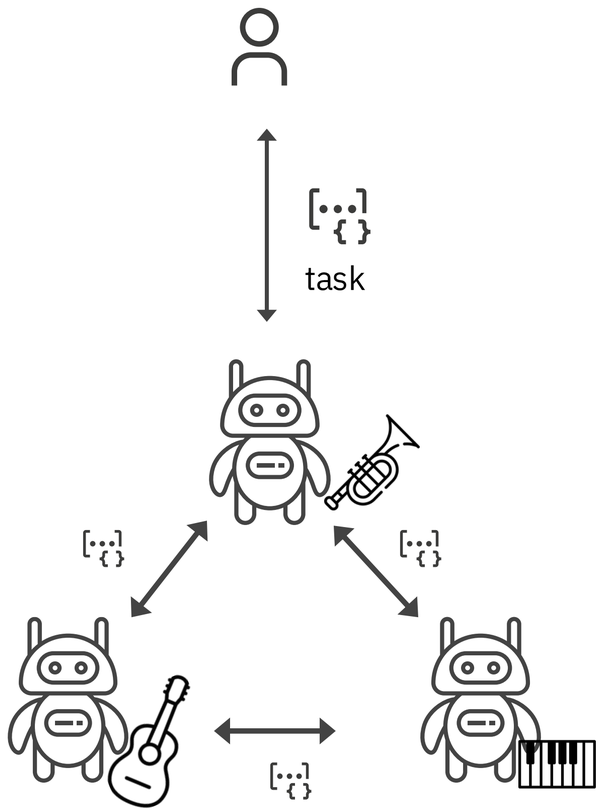
Multi-agent AI is powerful—but only with coordination, like musicians without a structure, a plan to follow agents cause chaos/confusion and not lead to expected results.

Giving a single LLM a prompt is already complex enough, now imagine:

- Each agent receives the same user query
- No rules, tools, no plan
- Outcome: conflicting, noisy responses

reference for the comparison made: link

Building Agentic System: Orchestra or Chaos?



Multi-agent AI is powerful—but only with coordination, like musicians without a structure, a plan to follow agents cause chaos/confusion and not lead to expected results.

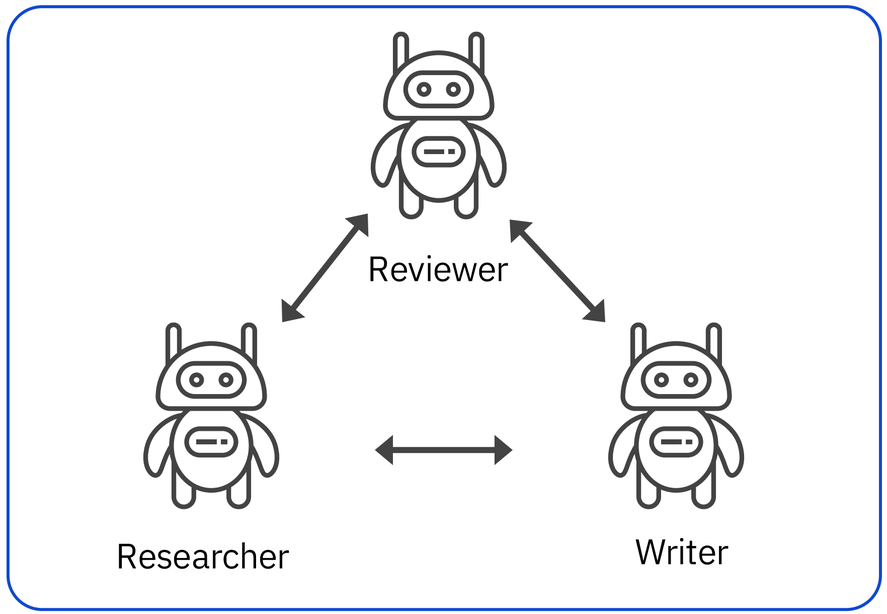
Now if we organize them and give each one a defined task:

- Each agent receives information
- Each has a defined tool and role
- Outcome: organized, efficient response

Structure, roles, and tools transform noise into harmony.

reference for the comparison made: link

Agent Communication and Orchestration in AutoGen

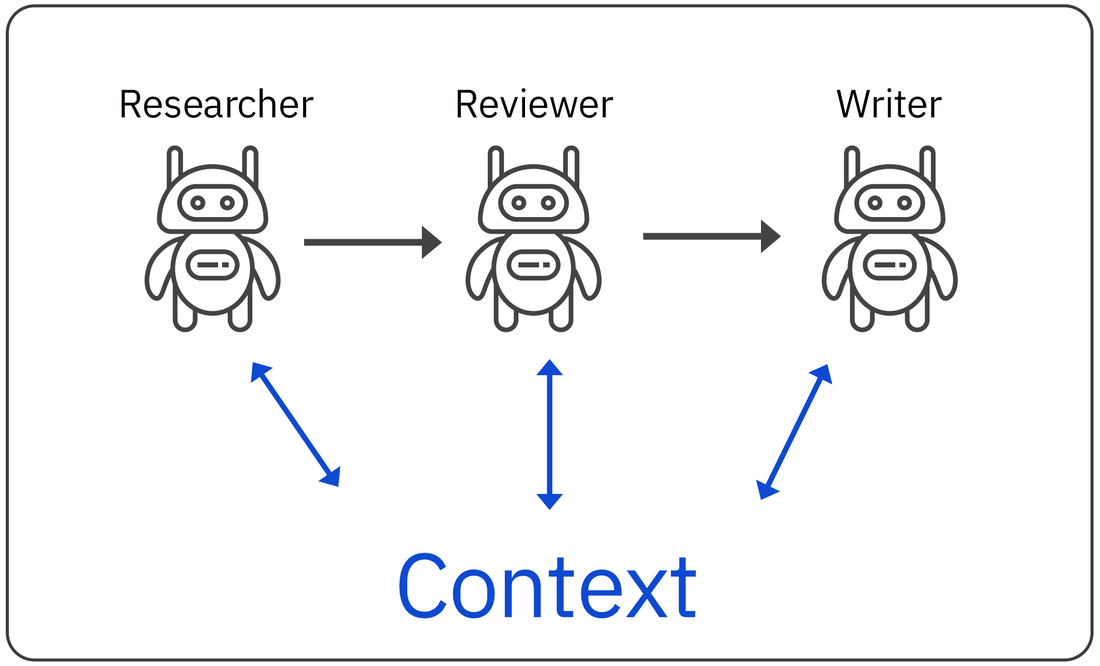


AutoGen manages agent communication using **group chat orchestration**.

This system allows agents to exchange messages in multi-turn conversations while keeping track of the context and state.

Basic Group Communication
Round-Robin Style

RoundRobinGroupChat



The most basic group structure in AutoGen is the RoundRobinGroupChat.

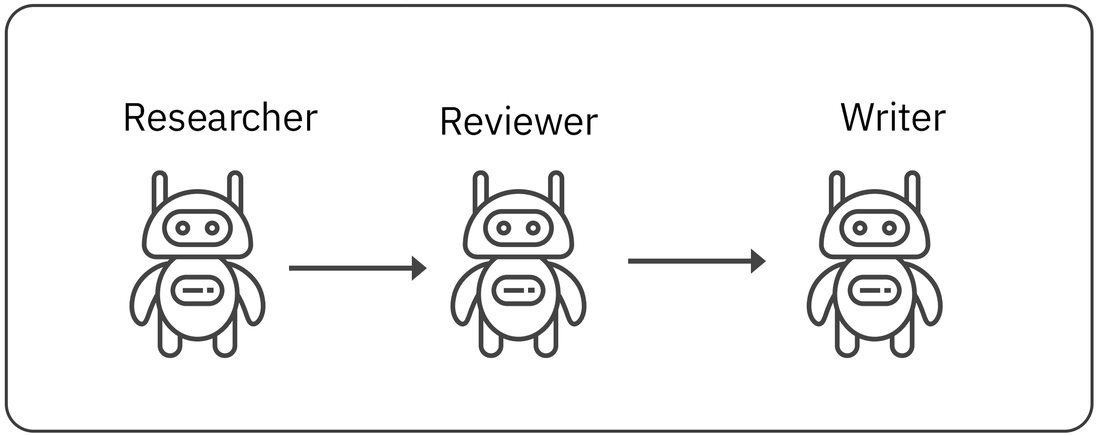
It’s a simple but powerful setup where all agents share the same context.

Each agent takes turns to speak, in a round-robin manner (in sequence).

When an agent responds, it broadcasts its message to all others, ensuring the whole group stays synchronized and aligned.

RoundRobinGroup configurations

RoundRobinGroupChat



RoundRobinGroupChat([Agent1, Agent2, Agent3])

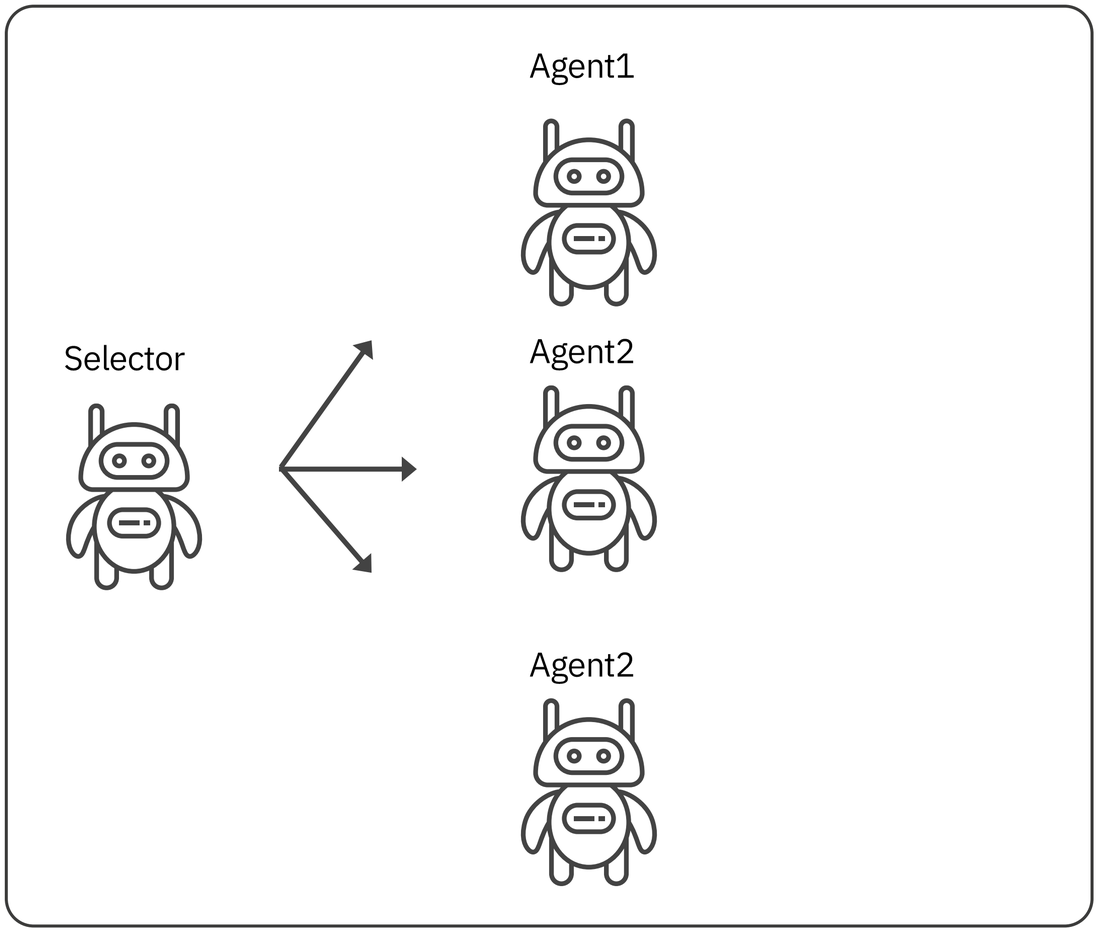
In a RoundRobinGroupChat, the order in which agents speak is defined when the group is created. You list the agents in the desired sequence, and they take turns accordingly.

Another important configuration is “stopping the conversation”.

You can control this using:

- max_turns: limits the number of turns
- termination_condition: a function that evaluates when to end the chat

SelectorGroupChat



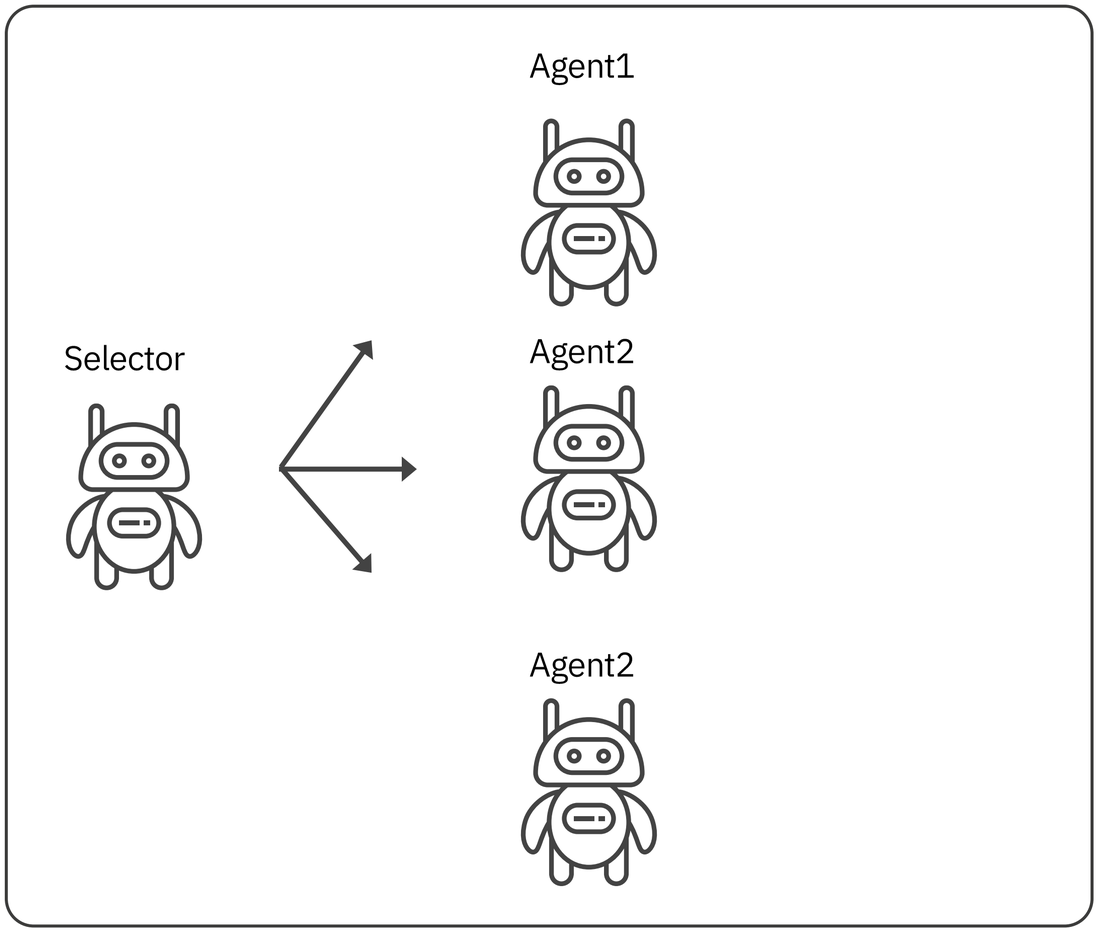
Another possible organization of group in autogen is the SelectorGroupChat, in this type of orchestration we will need one agent responsible for coordinating when each agent of the group will talk.

Now the decision of who talks is automated and will be made by an LLM.

Attention: This will give our workflow some randomness, because it is powered by LLMs to make the choice, hallucinations might be on our way

SelectorGroup Configurations

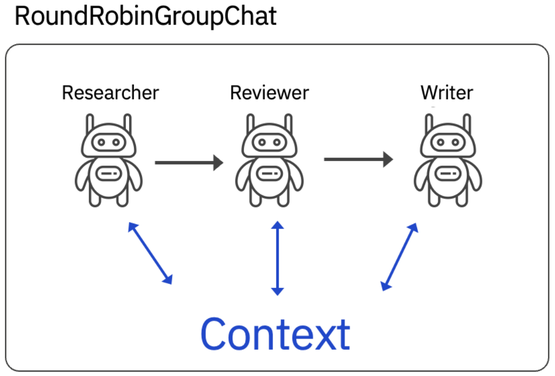
SelectorGroupChat



As the selection is made by an llm, we can still control how the choice is made by the prompt we give to the selector agent, defining the selector_prompt parameter.

You still need to configure when the system will stop by giving a termination_condition, it can be the maximum number of rounds, or it can be until the system reaches to an especific anwser of an especific agent.
Giving more autonomy to our system.

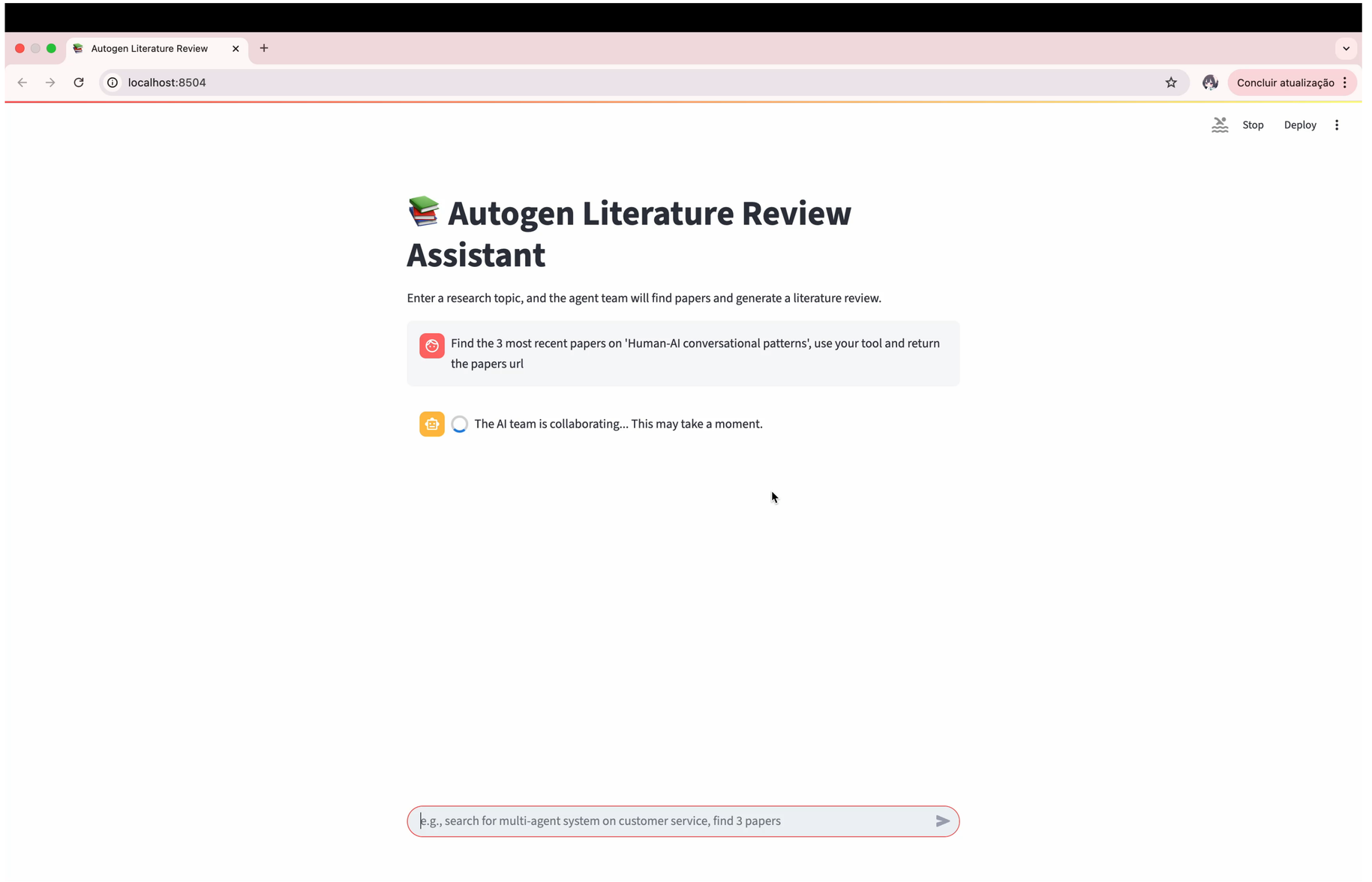
RoundRobinGroup configurations




```
    # Configuration of the team
    self.team = RoundRobinGroupChat([self.researcher, self.reviewer, self.writer], max_turns=3)
```

```
async def run_chat(self, task: str) -> str:
    """
    Runs the agent team and processes the event stream to build a clean log,
    correctly parsing all event data structures.
    """
    stream = self.team.run_stream(task=task)

    conversation_flow = [{"source": "User", "content": task, "type": "text"}]
```

CUI


Autogen Literature Review
localhost:8504
Concluir atualização
Stop
Deploy
📚 Autogen Literature Review Assistant
Enter a research topic, and the agent team will find papers and generate a literature review.
Find the 3 most recent papers on 'Human-AI conversational patterns', use your tool and return the papers url
The AI team is collaborating... This may take a moment.
e.g., search for multi-agent system on customer service, find 3 papers

Testing the workflow with an interface

Autogen-Conference

Setup Ollama

Install Ollama by following the instructions on the official website for your operating system. Link for download: https://ollama.com/

Download Models

Use the following commands to download the required models:

```
ollama pull granite3.3:2b
ollama pull granite3.3:8b
```

Setup Instructions - Run locally

1. **Create an virtual enviroment**

```
pip install virtualenv
virtualenv autogen_env

source sutogen_env/activate
```

2. **Install Autogen**

```
pip install pip install -U "autogen-agentchat"
pip install -U "autogen-ext[ollama]"
pip install streamlit
```

3. **Run locally - interface**

```
streamlit run main.py
```

link repository

Link pulbic github repository

Testing the workflow with an interface

link repository

Autogen-Conference

Setup Ollama

Install Ollama by following the instructions on the official website for your operating system. Link for download: https://ollama.com/

Download Models

Use the following commands to download the required models:

```
ollama pull granite3.3:2b
ollama pull granite3.3:8b
```

Setup Instructions - Run locally

1. **Create an virtual enviroment**

```
pip install virtualenv
virtualenv autogen_env

source sutogen_env/activate
```

2. **Install Autogen**

```
pip install pip install -U "autogen-agentchat"
pip install -U "autogen-ext[ollama]"
pip install streamlit
```

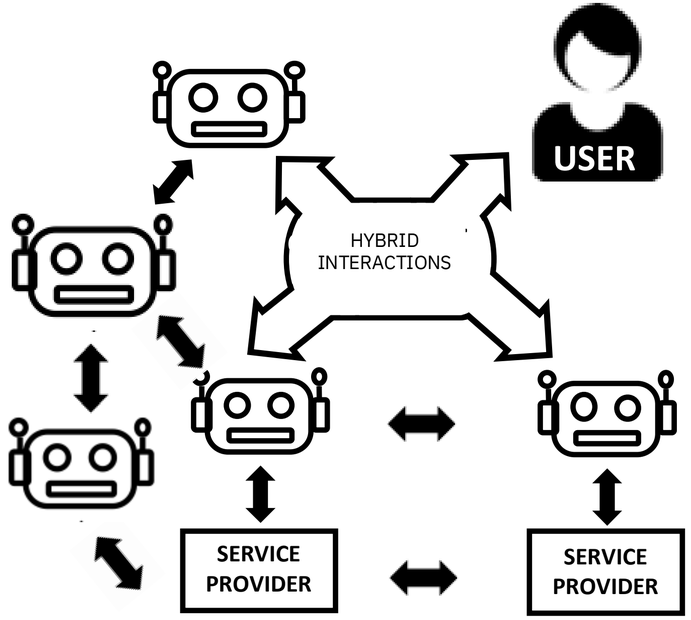
3. **Run locally - interface**

```
streamlit run main.py
```

Future Directions

Evolving directions

- Models are increasingly “agentic by default.” (o3, Claude 4, and DeepSeek- tool-calling capabilities and reasoning skills
- “Computer-use" agents that can directly manipulate GUIs (e.g., OpenAI Operator), as well as embedded agents within productivity tools like Notion or Copilot.
- Model Context Protocol (#MCP) and Agent-to-Agent (#A2A) communication standards (agent systems can collaborate across platforms)
- Agents with more autonomy and less delegation
- Agents operate as autonomous service providers



AGENTIC SYSTEMS

Rothschild, David M., et al. "The Agentic Economy." *arXiv preprint arXiv:2505.15799* (2025).

Zhu, Shenzhe, et al. "The Automated but Risky Game: Modeling Agent-to-Agent Negotiations and Transactions in Consumer Markets." *arXiv preprint arXiv:2506.00073* (2025).

Thank you!

Heloisa Candello – hcandello@br.ibm.com
Michelle Brachman – michelle.brachman@ibm.com
Amanda da Silveira - Amanda.da.Silveira@ibm.com