Agents - Patterns, Best Practices and Use Cases
Citigroup Talk

May-2025
Julian Kaljuvee @ Predictive Labs Ltd / Microsoft

Agenda

- Agents - Definition
- Agents - Evolution
- Agent Patterns and Best Practices
- Comparison of Popular Frameworks
- Use Case No 1 - Investment Committee
- Use Case No 2- Trade Execution Agent
- Risks and Mitigation Controls

Agents - Definition

There are multiple definitions of agents, but most assign AI agents the following features:

- **Autonomy** - software that can autonomously interact with their environment, make decisions, and take actions to achieve specific goals without constant human intervention (not fully true, we usually have human in the loop but the extent varies)
- **Memory and Tools** - equipped with reasoning capabilities, memory, and the ability to use tools or interact with external systems to complete complex tasks.
- **Planning, correction and adaption** - distinguished from traditional AI by their capacity for multi-step planning, self-correction, and adaptive behavior based on feedback from their actions.
- **Perception** - gathers data from its environment via sensors or inputs (eg especially in industry and robotics).

Agents - Evolution

Over the past few years Generative AI has evolved in the following steps:

1. **Basic LLM Workflow (2022)** - takes basic text / prompt as input, process handled solely by LLM, output text (no tools or memory)
2. **LLM with RAGs / Context and Tool Use (2023)** - retrieval-Augmented Generation (RAG) introduced, can call external tools (eg search engines, APIs), external data used to ground LLM outputs
3. **Multi-Modal LLM Workflow (2024)** - handles multi-modal input: text, image, audio, integrates both Tool Use and Memory, enables richer interactions across input types, improved reasoning via memory-assisted generations.
4. **Advanced AI Agent Architectures (2025+)** - improved tool acquisition via Model Contact Protocol (MCP), full agent orchestration layer and squads: dynamic task allocation, inter-agent communication, and monitoring, introduction of short-term, long-term, and episodic memory for more complex tasks.

Agents Architectures and Patterns

- **Single Agent System:**
  - LLM with tools and workflow
  - Connect tools to an AI Agent node, eg Mistral, Claude, OpenAI or Gemini
  - Add Simple Memory so it remembers your conversation
- **Multi Agent System:**
  - Distributed processing across specialized AI agents
  - Clearly assign agent roles and objectives (e.g., planner, researcher, reviewer).
  - Scalable architecture that can grow with complexity
  - Parallel execution for improved performance on complex tasks
  - Ideal for cross-domain problems requiring multiple types of expertise
  - In trading, this could be a team or squad of signal generation, trading execution and reporting agent

Agents Architectures and Patterns (continued)

- **Popular Patterns:**
    - **Sequential** - chain agents together like an assembly line
    - **Parallel** - run multiple agents simultaneously (performance optimisation)
    - **Loop** - create iterative improvement cycles (good for content creation)
    - **Router** - use a switch node to direct flow based on input type
    - **Aggregator** - combine inputs from multiple sources into one agent
    - **Network** - create a mesh of interconnected agents
    - **Hierarchical** - manager/worker relationships between agents
- **Specialised Patterns:**
    - **Human-in-the-loop** - get human approvals for important steps
    - **Shared tools** - multiple agents accessing Vector Search and Web Search
    - **Database with tools** - connecting directly to databases
    - **Memory transformation** - using custom tools to enhance memory

Agents Patterns - Best Practices

- **Planning** - ask the agent to plan its work first, then execute, then see the response
- **Add memory to agent** - otherwise your agent is basically a goldfish
- **Use loops for any complex processes** - drag that output right back to the input
- **Tell the agent which order to use tools** - like "first search, then summarize"
- **Give your tools super clear descriptions** - consistent responses
- **Always ask "Return Intermediate Steps"** - lifesaver for debugging
- **Use consistent naming in your workflows** - future you will be grateful
- **Assign clear roles to each agent** - just like you would with a human team

Comparison of Popular Agentic Frameworks

Framework	Key Strengths	Ideal Use Cases	Learning Curve	Notable Features
LangGraph	Graph-based orchestration for complex, stateful workflows.	Multi-step reasoning, decision trees, research assistants.	Moderate	Visual workflow design, state management, LangChain integration.
AutoGen	Conversational agents with flexible collaboration and human-in-the-loop capabilities.	Coding assistants, simulations, collaborative agents (Microsoft)	Moderate	Modular agents, dynamic conversations, tool integration.
CrewAI	Role-based multi-agent systems with intuitive setup.	Task delegation, team-based workflows, rapid prototyping.	Low	YAML configuration, CLI tools, role assignments.
OpenAI Agents SDK	Lightweight, production-ready framework with minimal abstractions.	Task automation, agent delegation, secure workflows.	Low	Agents, handoffs, guardrails, tracing, Python-first design.
Google Agent Development Toolkit (ADK)	Model Agnostic, multi language	Multi agentic workflows	Moderate	Pre-built tools (Google Search, Code Execution, MCP)
SmolAgents	Lightweight framework for building simple AI agents.	Quick prototyping, educational purposes, lightweight applications (HuggingFace)	Low	Minimal setup, easy customization, rapid deployment.

Use Case 1 - AI Investment Committee Advisor

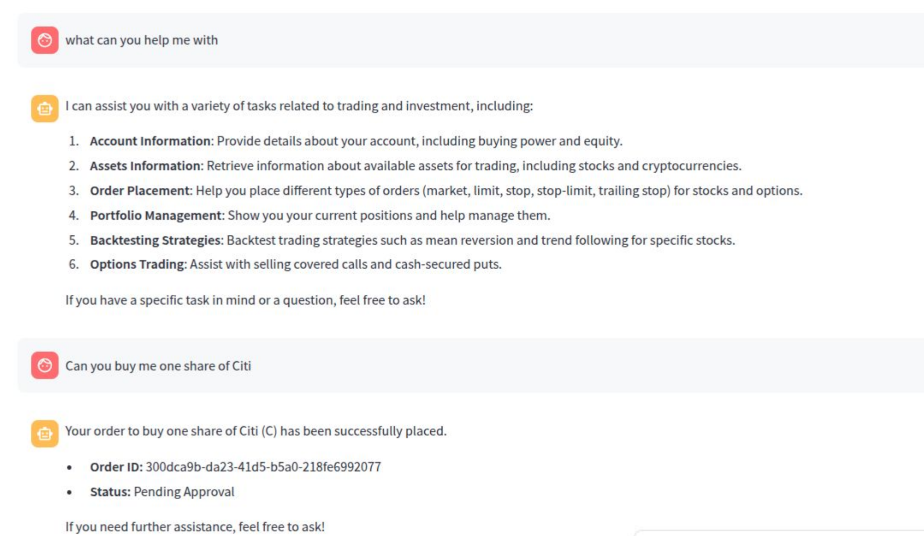
Building ‘AI investment committee advisor’ for a large UAE client with the following features:

- **Core System** - multi-agent framework with specialized sub-agents for research, risk assessment, and document analysis, built on foundation models with function calling capabilities.
- **Data Integration** - connect internal portfolio systems and external market data (Bloomberg, SEC filings, news feeds) through secure APIs with real-time updates.
- **Chat Interface** - natural language queries with conversational memory, multi-user collaboration, and proactive insights triggered by new investment opportunities.
- **Deal flow generation and due diligence** - automated research packages for new opportunities, comparative analysis against portfolio holdings, scenario modeling with stress testing, and regulatory compliance monitoring.
- **Human-Centric Design** - augments committee judgment with comprehensive research and pattern recognition while maintaining human oversight for final investment decisions and audit trails for compliance (agent can be a non-voting member of the committee).

User Case 2 - Streaming Trade Execution Agent with LangGraph

- **What** - a simple agent built with LangGraph that can get account information and execute trades on Alpaca Markets
- **Why** - in about 6-12 months time, all major brokerages (eg Lightyear, Robinhood, IBKR, Citigroup) will have this type of chat interface and most likely will also support also voice.
- **How** - the agent has the following building blocks:
  - **Alpaca Markets API** - you give the agent access to Alpaca Market API Keys
  - **State Management** - create a state object that tracks the state of the agent
  - **Tools** - create a series of tools, such as as get_account_info, get_assets, get_positions and place_market_order and attach them to a ToolNode
  - **Workflow -** create a graph with edges and notes representing the workflow: User Input → 2. Agent Processing → 3. Tool Selection → 4. API Execution → 5. Response Formatting
  - **Memory** - conversation history and threading
  - **Debugging (LangSmith)** - features built-in tracing and visualization tools to inspect the execution path and state changes during runtime
- **Stack** - LangGraph, LangChain, FastAPI, and Streamlit / Vercel for visualisation
- **Demo -** a demo on Streamlit or Zuvu.ai Marketplace
- **Source** - there are additional tutorials on LangChain AI LangGraph Github

LangGraph AlpacaTrader - Running It



what can you help me with

I can assist you with a variety of tasks related to trading and investment, including:

1. **Account Information**: Provide details about your account, including buying power and equity.
2. **Assets Information**: Retrieve information about available assets for trading, including stocks and cryptocurrencies.
3. **Order Placement**: Help you place different types of orders (market, limit, stop, stop-limit, trailing stop) for stocks and options.
4. **Portfolio Management**: Show you your current positions and help manage them.
5. **Backtesting Strategies**: Backtest trading strategies such as mean reversion and trend following for specific stocks.
6. **Options Trading**: Assist with selling covered calls and cash-secured puts.

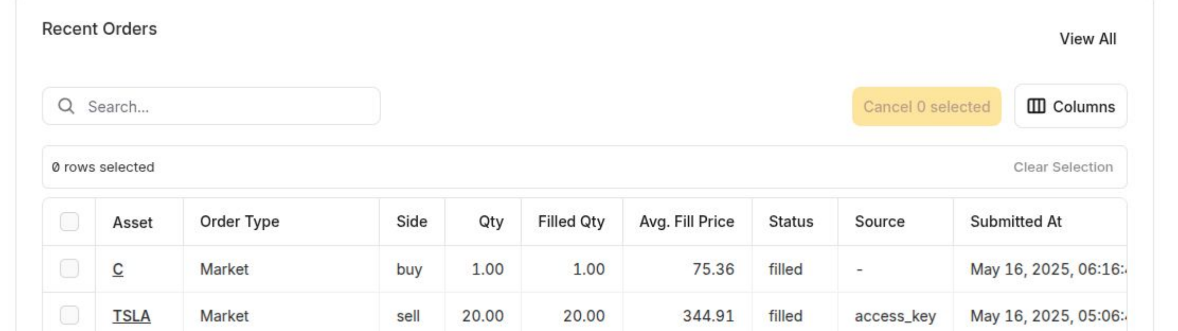
If you have a specific task in mind or a question, feel free to ask!

Can you buy me one share of Citi

Your order to buy one share of Citi (C) has been successfully placed.

- **Order ID:** 300dca9b-da23-41d5-b5a0-218fe6992077
- **Status:** Pending Approval

If you need further assistance, feel free to ask!



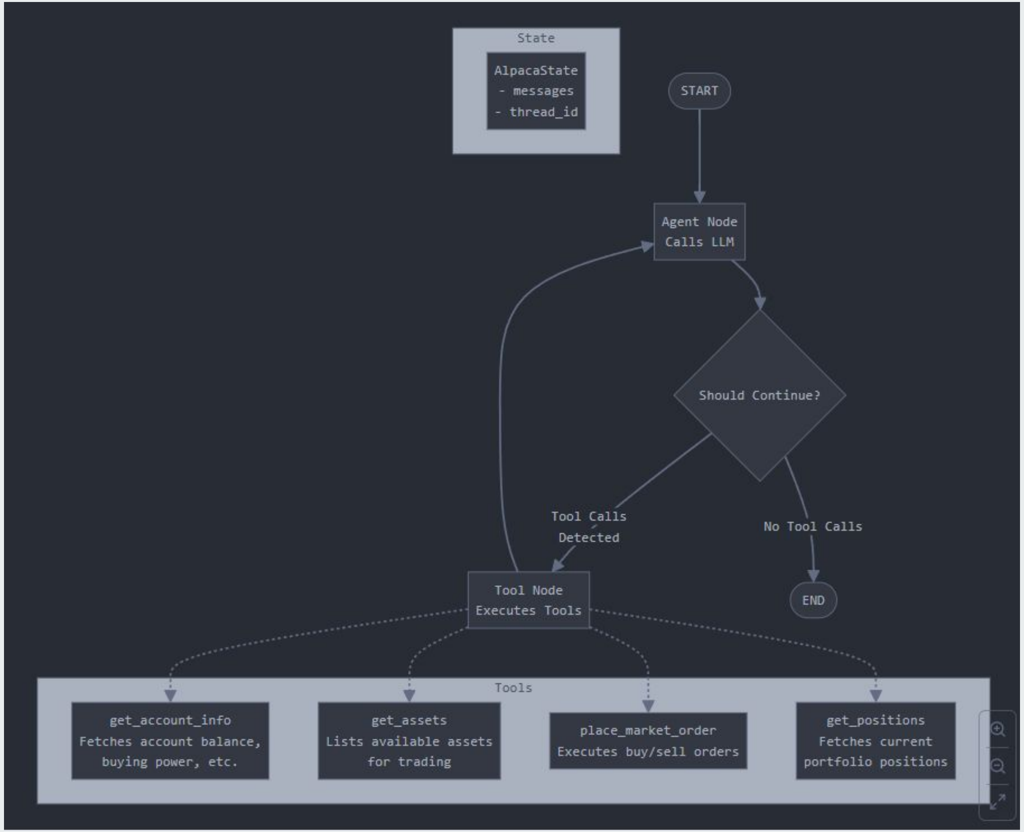
Recent Orders

0 rows selected

☐	Asset	Order Type	Side	Qty	Filled Qty	Avg. Fill Price	Status	Source	Submitted At
☐	C	Market	buy	1.00	1.00	75.36	filled	-	May 16, 2025, 06:16:
☐	TSLA	Market	sell	20.00	20.00	344.91	filled	access_key	May 16, 2025, 05:06:

Sources: Streamlit and Zuvu.ai

LangGraph Alpaca Trader - Graph Visualization



Agents - Risk Mitigation and Controls

Given the inherent non-deterministic nature of generative AI agents, there will be increased focus on both key risks and controls, especially in financials services:

Key Risks

- **Regulatory and compliance risks** - model bias in lending decisions, lack of explainability for regulatory scrutiny, and potential violations of consumer protection laws
- **Operational risks** - model drift degrading performance over time, data quality issues affecting decision accuracy, and cybersecurity vulnerabilities in AI systems
- **Reputational and financial risks** - algorithmic discrimination, customer privacy breaches, and systematic errors in high-stakes financial decisions

Mitigation Controls

- **Validation and governance** - robust model governance with regular bias testing, model validation frameworks, continuous monitoring dashboards, and clear escalation procedures for performance degradation
- **Risk management infrastructure** - model risk committees, independent validation teams, comprehensive documentation standards, and regular third-party audits
- **Technology safeguards** - data encryption, access controls, model versioning, rollback capabilities, and human oversight requirements for high-impact decisions

Resources

- **Alpaca Markets** – a leading API-first broker (supports paper trading)
- **LangGraph** – popular open-source agent framework by LangChain
- **OpenAI Agent Guide** - a practical guide to building agents
- **Google Agent Development Kit** - ADK or Agent Development Kit from Google
- **LangChain Open Canvas** - open source UI to connect to agents
- **Anthropic MCP** - a new standard for connecting AI assistants to the systems where data lives
- **Trade Execution Agent Github Repo** - open source repo for the trade execution agent, live front end at Streamlit.io
- **Zuvu.AI - ** agent marketplace the trade execution demo
- **Hugginface Smol Agents** - open source platform for integrating different LLMs