FLORIDA ATLANTIC

COT 6930 - Generative Intelligence
and Software Development Lifecycle

Topic 6 -
Agentic AI Systems

Dr. Fernando Koch
kochf@fau.edu
http://www.fernandokoch.me

Agenda

Foundations of Agentic AI

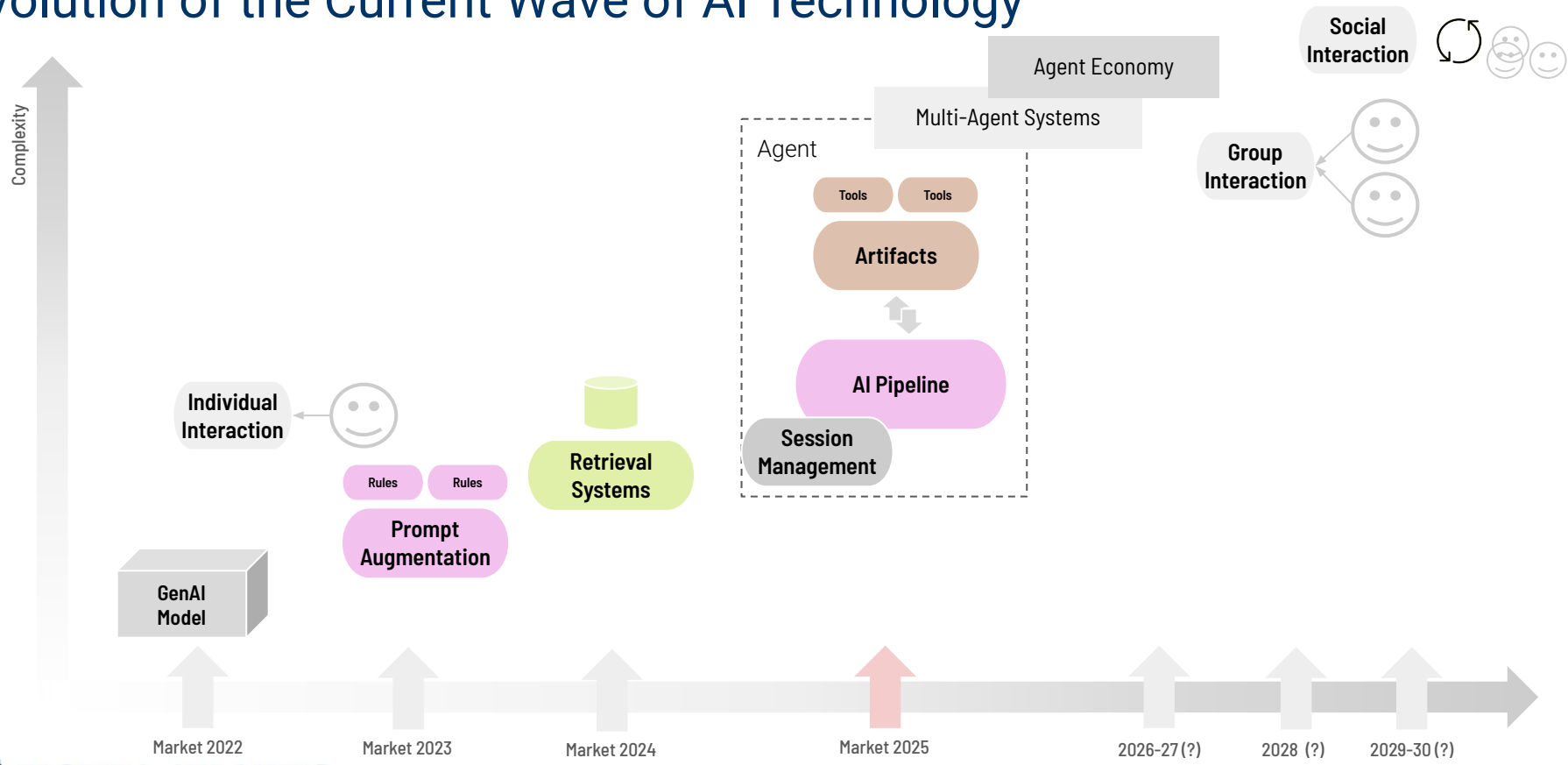
Agentic Decision-Making and Task Delegation

Architectures and Tools

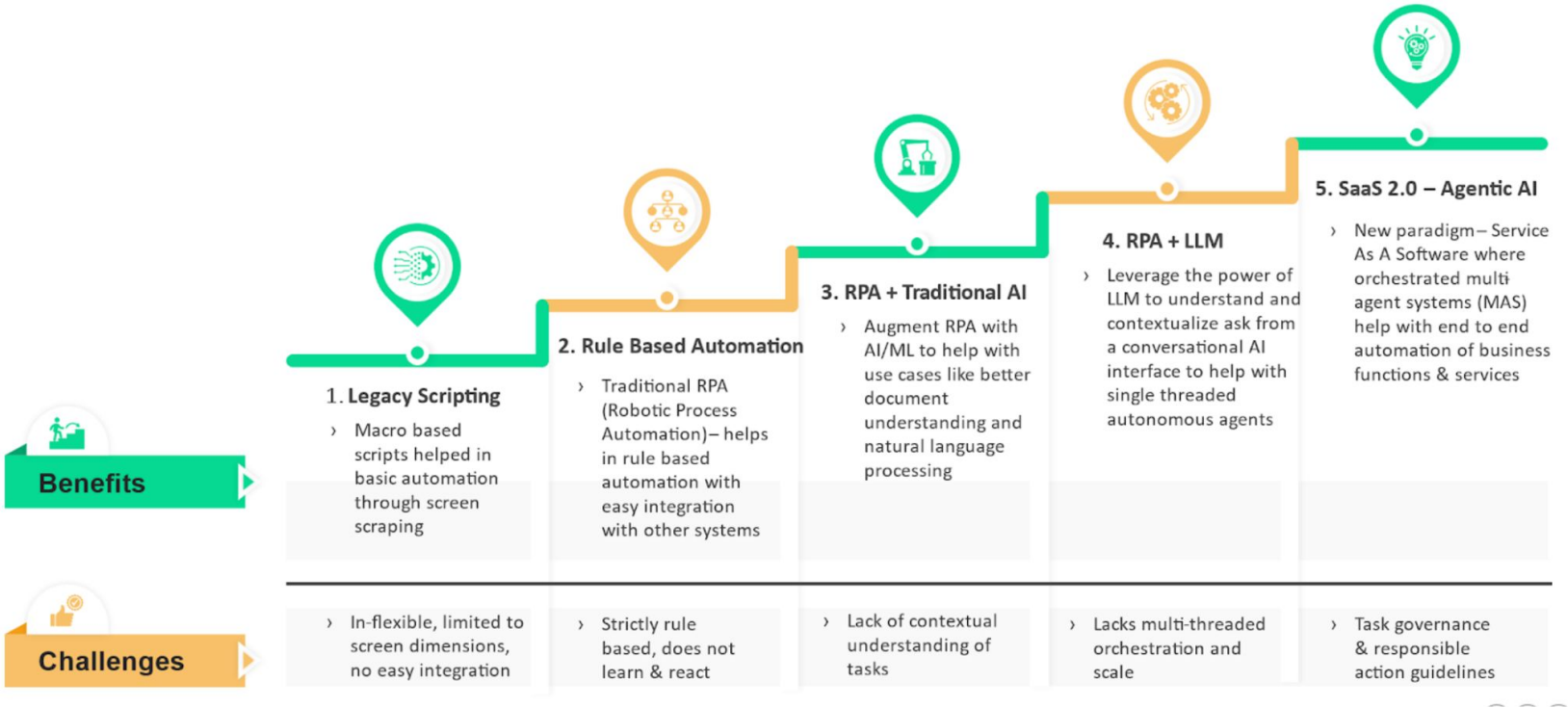
Our Key Question

How can we design intelligent, autonomous agents that make decisions, interact with environments, and coordinate tasks in software systems?

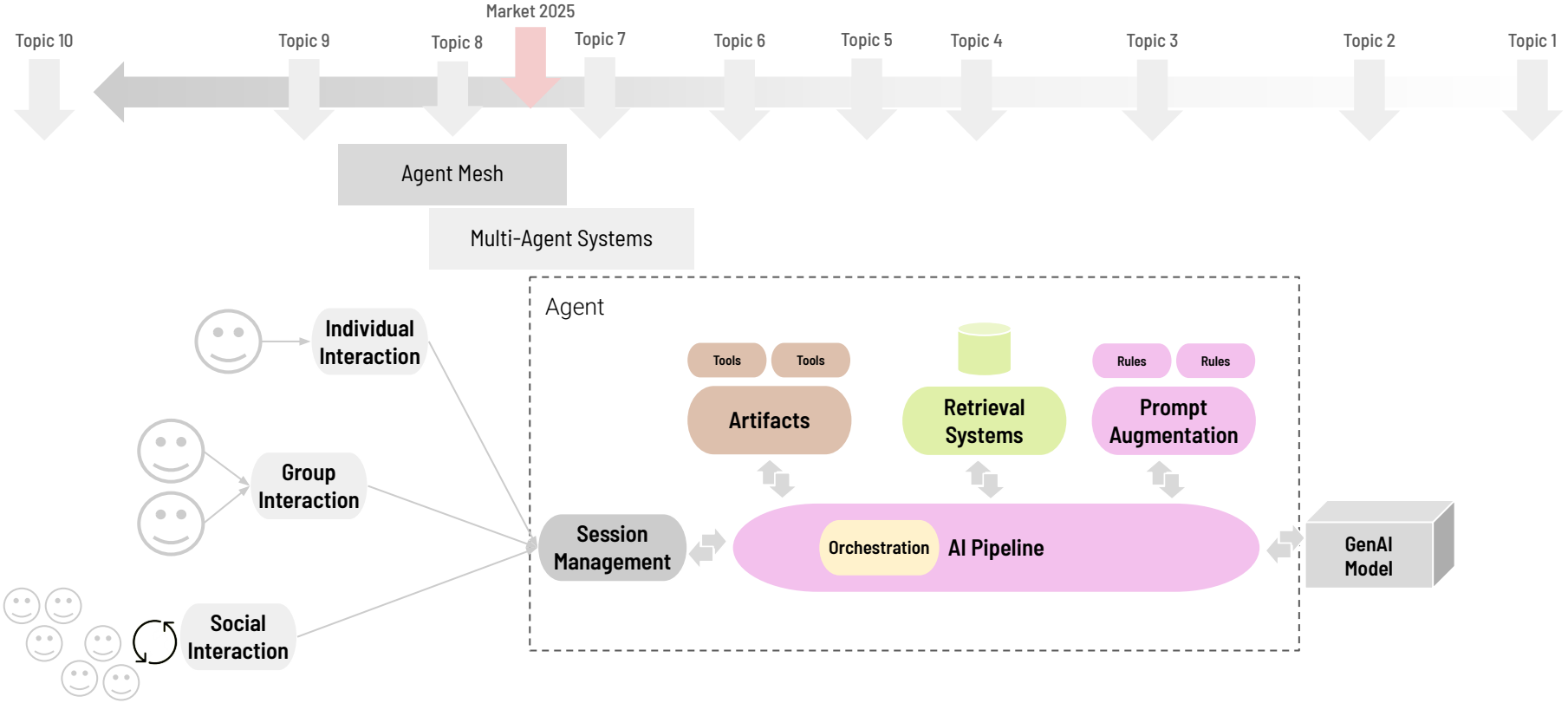
Evolution of the Current Wave of AI Technology



Evolution of Agentic Systems



Evolution of This Course

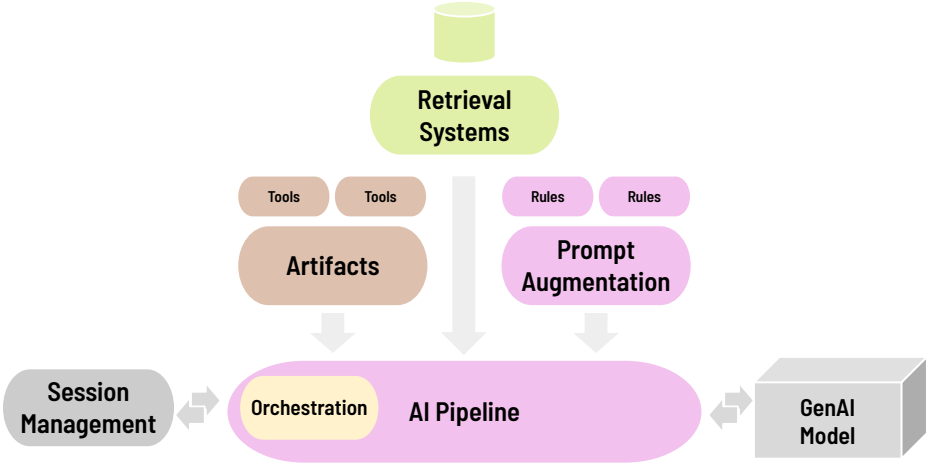


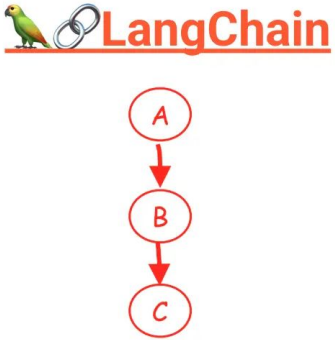
What are Orchestration Workflows?

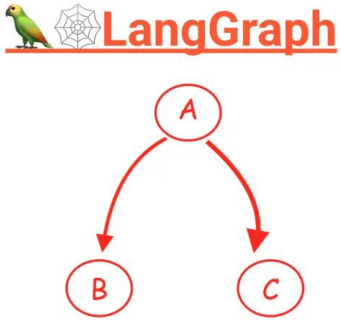
Frameworks to manage AI pipeline execution.

- Schedule, route, and monitor tasks
- Enable conditional branching and looping
- Support multi-step dependencies

- **LangChain**: modular chaining of model calls
- **LangGraph**: graph-based orchestration for adaptive flows



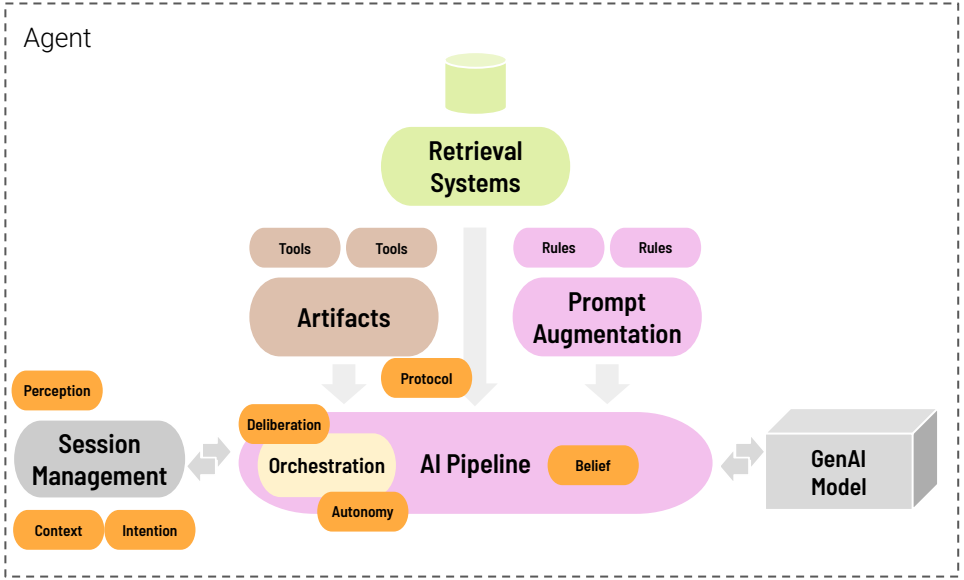




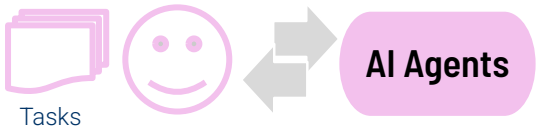
What are Agentic Systems?

Agentic systems =
LLMs + tools + memory + autonomy.

- Perceptions, Beliefs, Intentions.
- Tools
- LLM-driven components capable of planning



Levels of Automation in Agentic Systems



Level 1 - Human + AI
AI Agent augment human capacity to operate a task

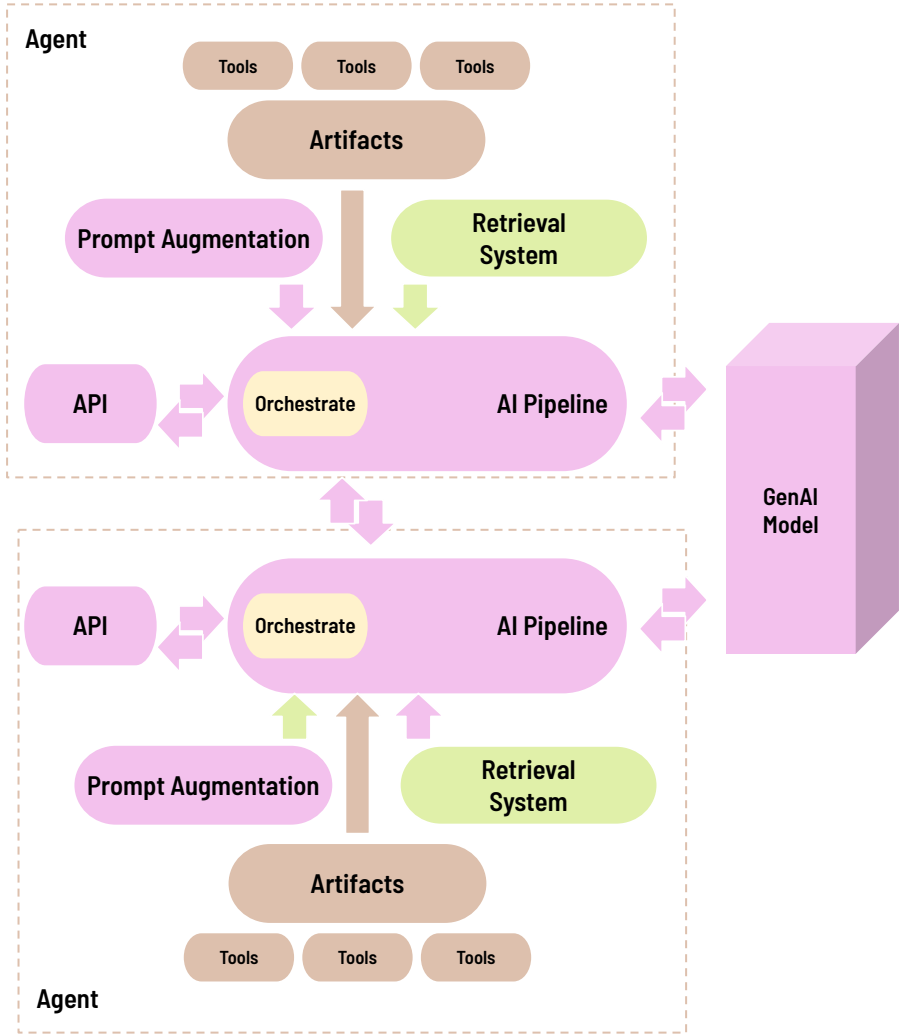
Level 2 - AI + Human
AI Agent implement tasks with Human-In-The-Loop task support

Level 3 - AI + AI w/ Human Oversight
Group of AI Agents implement tasks with Human-In-The-Loop task observation

What are Multi-Agent Systems?

Composed of multiple autonomous agents that collaborate, delegate, or compete to accomplish complex tasks. Each agent:

- Has its own goal, memory, tools, and orchestration logic.
- May perform specialized roles (e.g., planner, coder, tester, executor).
- Can communicate with other agents via shared context or message passing.

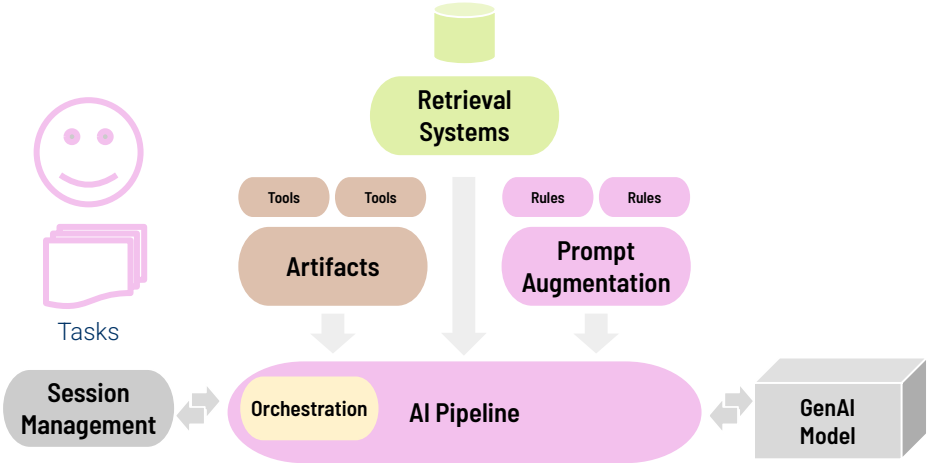


Orchestration

Answer the question: 'What to do next?'

Coordinates how the components interact to produce coherent results.

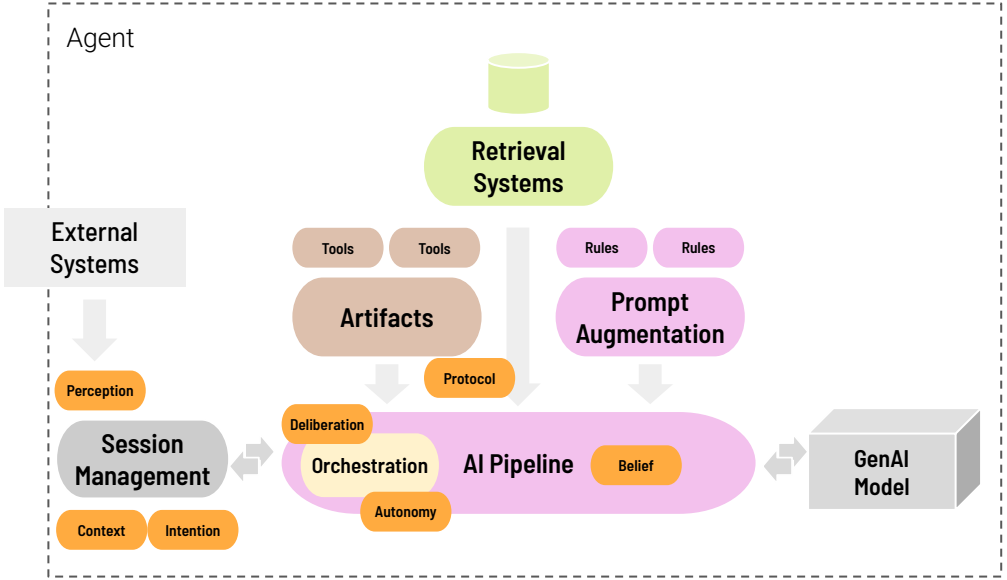
- **Flow Control:** Directs prompt requests through augmentation, retrieval, and response stages.
- **Adaptive Routing:** Dynamically chooses which systems (retrieval, tools, models) to invoke.
- **Error Handling & Validation:** Ensures robustness by refining or retrying when needed.



Perceptions

Allows agents to sense and interpret digital signals to understand their environment.

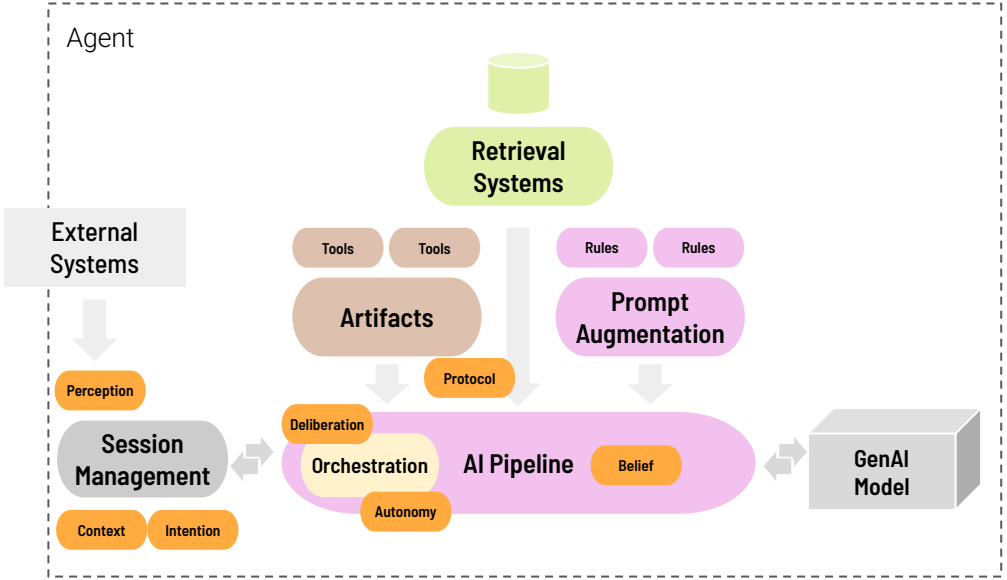
- Collects data from APIs, logs, databases, or user interfaces.
- **Converts raw inputs into structured representations (Sessions, Memory).**
- Forms the foundation for intelligent decision-making.



Beliefs

Belief-state models store the agent's internal understanding of its environment and goals.

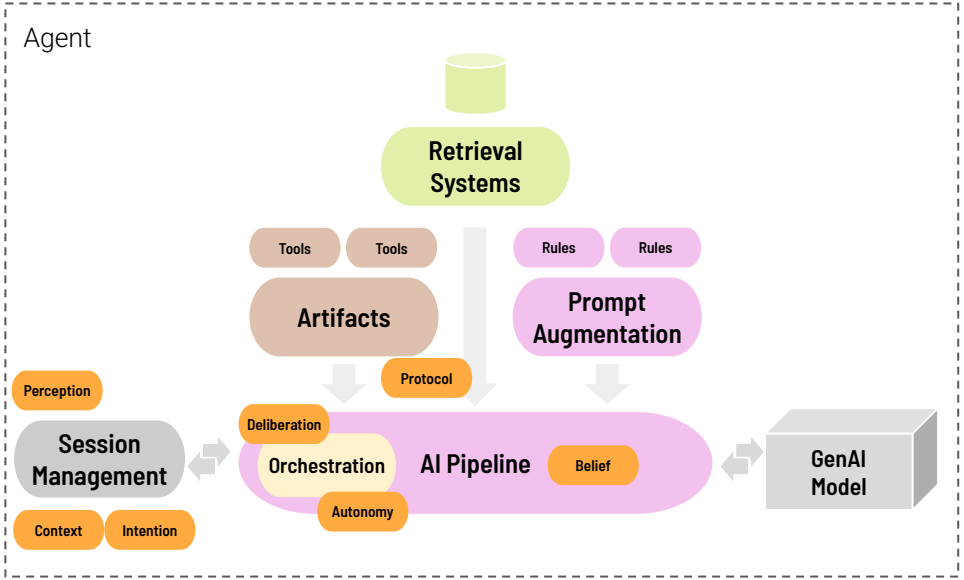
- Forms the "mental model" that guides autonomous actions.
- Represents current knowledge, assumptions, and uncertainties.
- Continuously updated as new information is perceived.
- **Enables reasoning under incomplete or evolving conditions.**



Deliberation

Agents autonomously decide what actions to take to achieve their goals within dynamic environments.

- Combines perception, reasoning, and planning to act intelligently.
- Uses goal-based or policy-driven planning methods.
- Balances short-term actions with long-term objectives.
- Enables goal-oriented behavior across complex software systems.



Architectures and Tools

Building Agents

How to build an agent

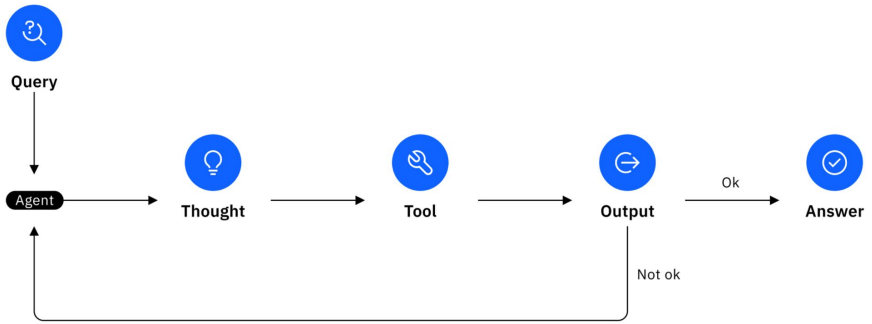
Building an agent is a process of designing workflows and connecting pieces of the OpenAI platform to meet your goals. Agent Builder brings all these primitives into one UI.

GOAL	WHAT TO USE	DESCRIPTION
Build an agent workflow	Agent Builder	Visual canvas for creating agent workflows. Brings models, tools, knowledge, and logic all into one place.
Connect to LLMs	OpenAI models	Core intelligence capable of reasoning, making decisions, and processing data. Select your model in Agent Builder.
Equip your agent	Tools, guardrails	Access to third-party services with connectors and MCP, search vector stores, and prevent misuse.
Provide knowledge and memory	Vector stores, file search, embeddings	External and persistent knowledge for more relevant information for your use case, hosted by OpenAI.
Add control-flow logic	Logic nodes	Custom logic for how agents work together, handle conditions, and route to other agents.
Write your own code	Agents SDK	Build agentic applications, with tools and orchestration, instead of using Agent Builder as the backend.

ReACT Agents

A ReAct agent combines reasoning ("think") and acting ("do") within a single AI framework, letting large language models plan and execute tasks dynamically.

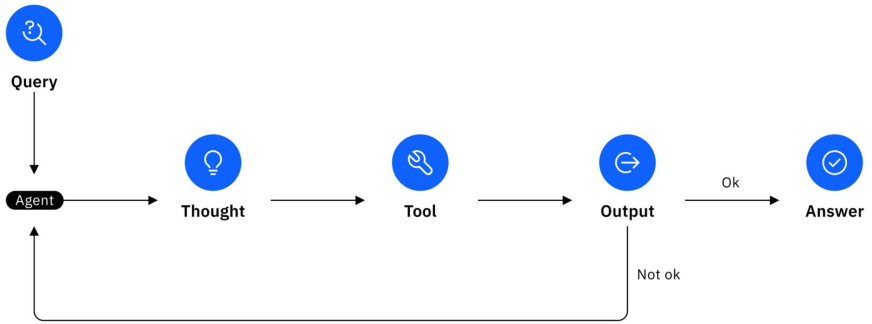
- Yao et al. (2023) in "ReAct: Synergizing Reasoning and Acting in Language Models."
- Integrates chain-of-thought (CoT) reasoning with tool use and actions.
- Enables LLMs to plan, execute, and adapt in real time instead of following static rules.
- Marks a shift from chatbots to truly agentic problem-solving systems.



Why do ReACT Agents matter?

The ReAct framework empowers AI agents to reason, learn, and adapt, forming the core of modern autonomous systems.

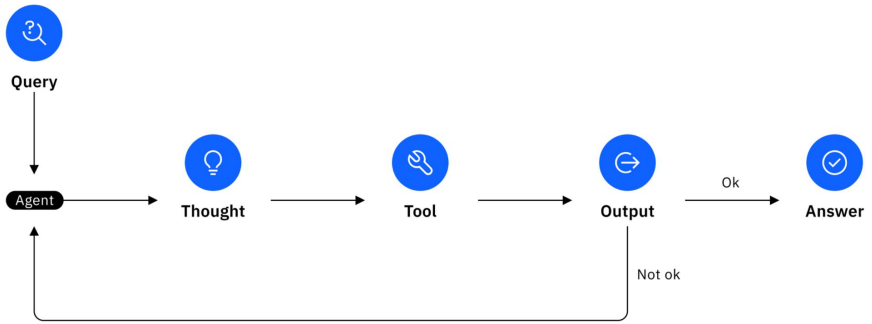
- Merges planning + execution in a continuous feedback loop.
- Allows dynamic task adaptation to new data or failures.
- Forms the basis for multi-agent workflows and tool-using agents (LangChain, AutoGen).
- Scales by delegating subtasks to smaller agents within a ReAct ecosystem.
-



How do ReACT Agents work?

Analogy: Like a human thinking aloud and adjusting steps after checking results.

Iterative loop of Thought → Action → Observation, using language-based reasoning to guide tool use and decisions.



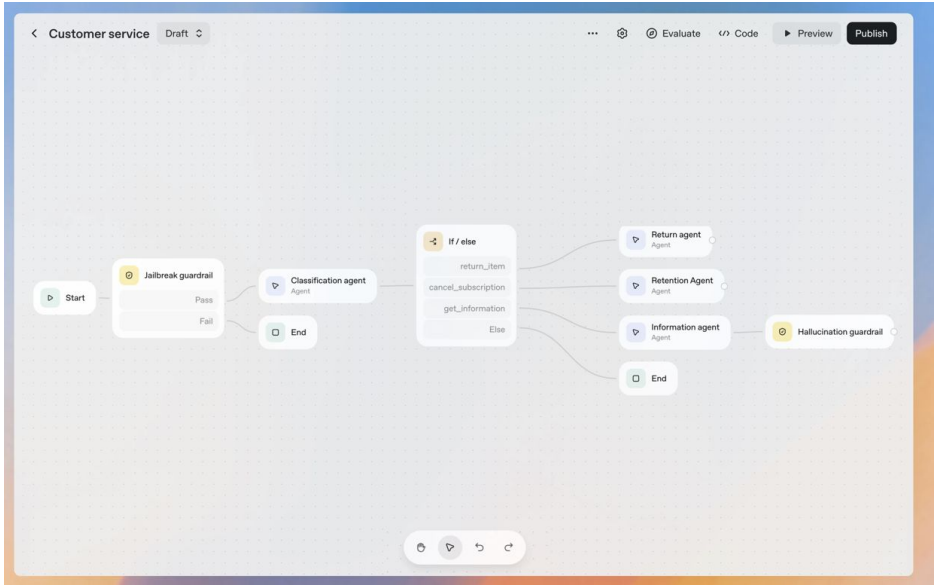
- The LLM verbalizes its reasoning (“thoughts”) to decompose a task.
- Executes an action (API call, search, calculation, etc.).
- Makes an observation from the result and updates its belief state.
- The loop repeats until the goal is met or no further actions are needed.

OpenAI AgentKit

https://openai.com/index/introducing-agentkit/

Set of tools for developers and enterprises to build, deploy, and optimize agents.

- **Agent Builder**: a visual canvas for creating and versioning multi-agent workflows
- **Connector Registry:** a central place for admins to manage how data and tools connect across OpenAI products
- **ChatKit:** a toolkit for embedding customizable chat-based agent experiences in your product

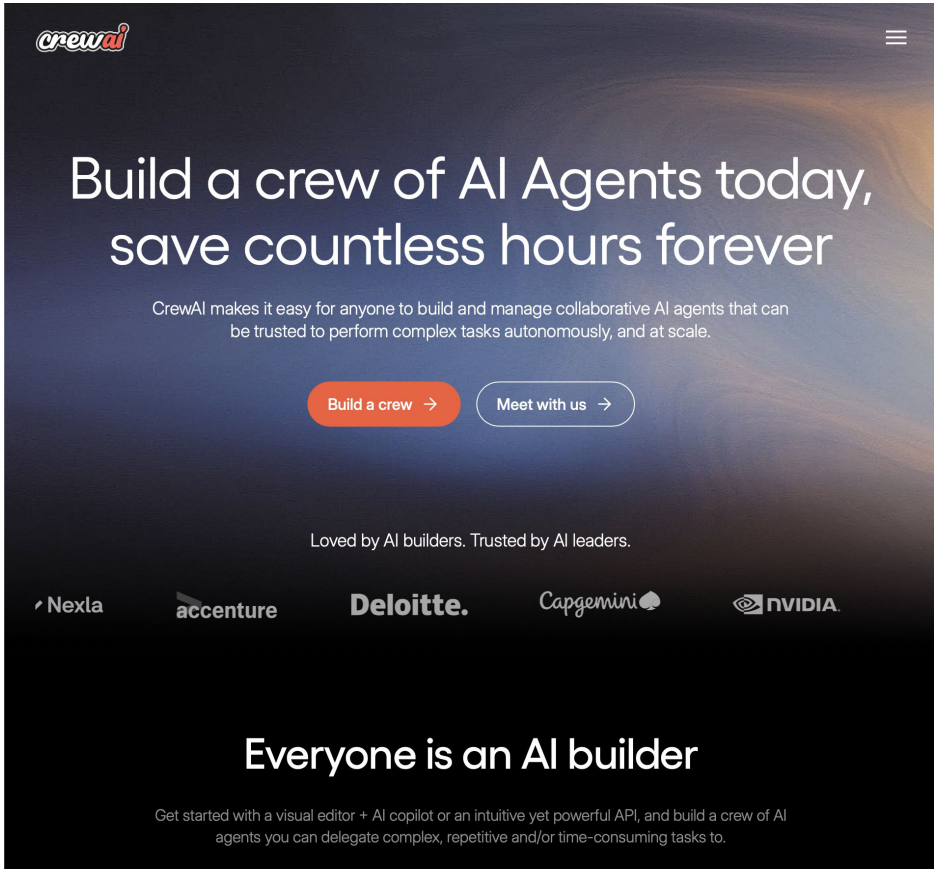


CrewAI

https://www.crewai.com

Multi-agent collaboration and orchestration framework.

- Enables teams of autonomous agents to work toward shared goals.
- Role-based agent design (e.g., researcher, coder, reviewer).
- Integrates with OpenAI, Anthropic, Hugging Face APIs.
- Supports task delegation, tool usage, and dialogue between agents.

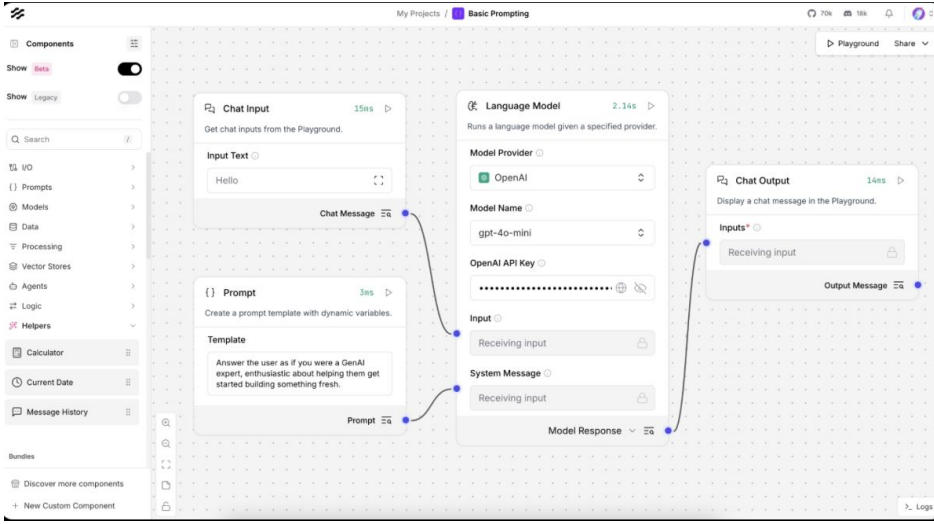


LangFlow

https://www.crewai.com

Visual low-code builder for LLM and agent workflows.

- Flow-based interface for building complex reasoning pipelines.
- Built on top of LangChain, compatible with OpenAI and local models.
- Drag-and-drop logic nodes for chaining perception, reasoning, and action steps.



Top Alternatives (2025)...

- **n8n:** A low-code automation platform that supports building agentic workflows, offering both open-source and commercial versions.
- **FlowiseAI:** Type: Open-source visual builder for LLM applications.
- **AutoGen:** A framework for building and orchestrating multi-agent conversational applications.
- **Zapier**: A widely used automation tool that is expanding its capabilities to include AI agents.
- **SmythOS:** platform that emphasizing multi-agent collaboration and deployment flexibility
- **Dify.AI:** Type: Full-stack AI app and agent orchestration platform.

EXERCISE

Exercise 6 - Agentic AI Hands-On Experiment

Objective:
Develop a basic Agentic AI with Tool Calling.

Go to Canvas -> Assignments

COT 6930 - Generative Intelligence and Software Development Lifecycles

Dr. Fernando Koch
kochf@fau.edu
http://www.fernandokoch.me