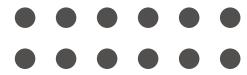




Common Agentic Design pattern - Part 1

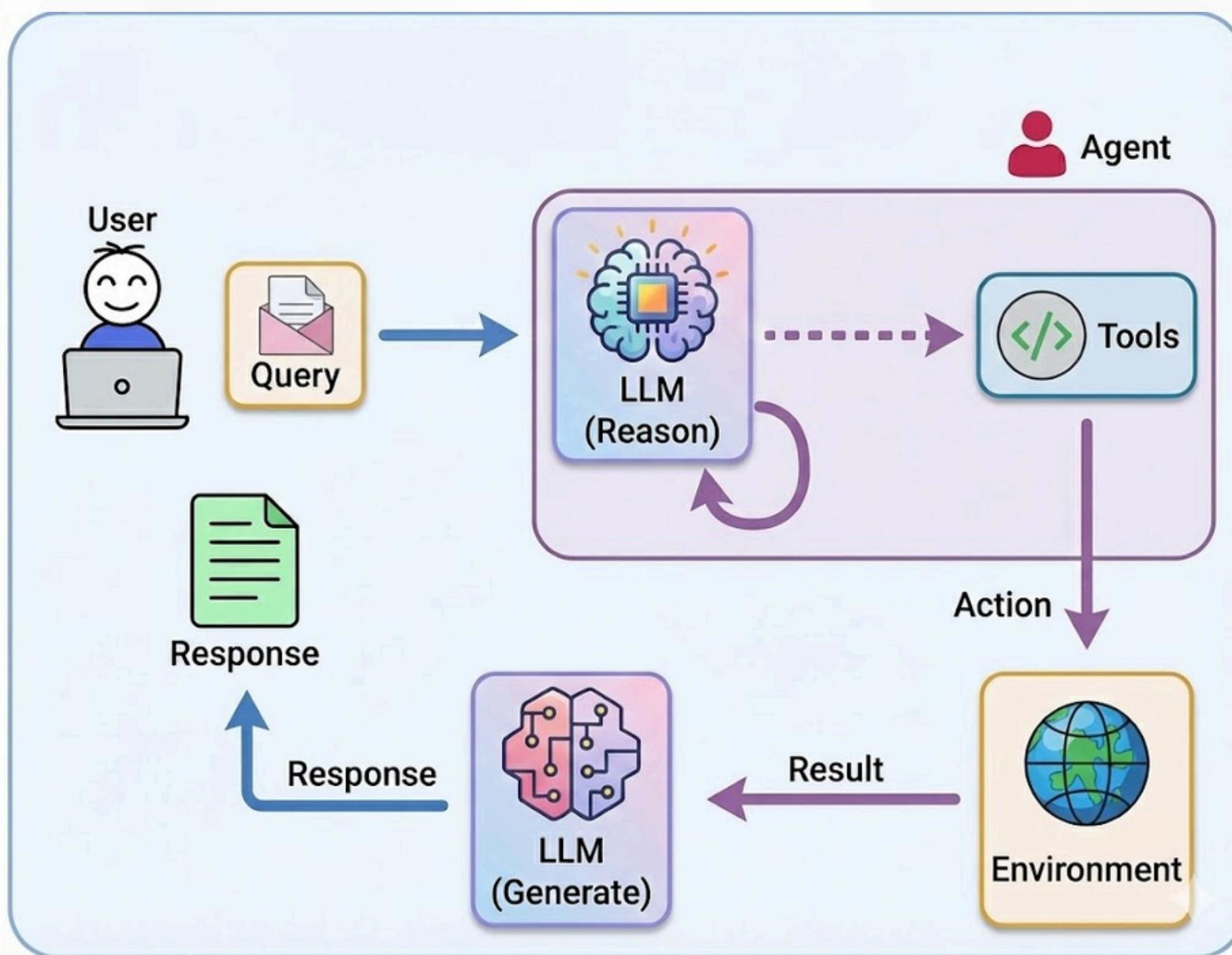
Check out AnanthaNarayanan

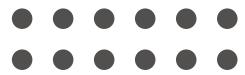


ReACT – Reasoning & Acting

A **ReAct** (Reasoning and Acting) agent is an AI system that solves complex tasks by combining step-by-step reasoning with external tool use.

It operates in a loop of **Thought** → **Action** → **Observation**, where it thinks through the problem, performs an action using a tool (e.g., search, calculator, API), observes the result, and repeats this cycle until it can deliver a final answer. This iterative process allows the agent to dynamically refine its plan, gather new information, and interact with external environments to solve problems effectively.



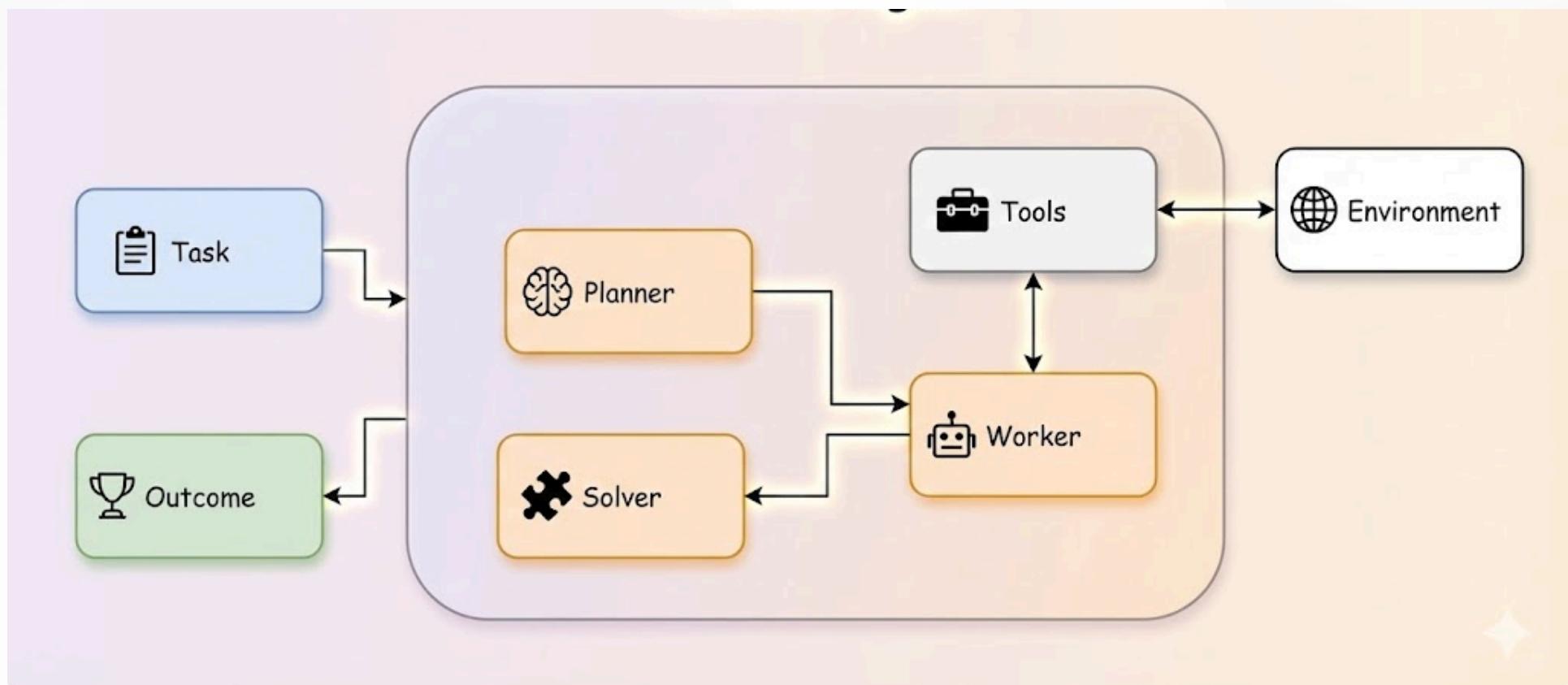


ReWOO – Reasoning without Observation

A **ReWOO** agent is an AI agent architecture where the model plans its **entire reasoning and tool-use sequence upfront**, then executes that plan, and only afterwards synthesizes the final answer.

ReWOO Agent proposed as a way to make augmented language-model agents more efficient. Instead of interleaving “think → call tool → see result → think again” at every step (as in **ReAct**), the agent first generates a full plan with placeholders for evidence, then fills those placeholders by calling tools, and finally uses the collected evidence to answer.

LLMs don’t always need to “wait and see”. They can “think ahead”, reducing redundant prompts and making AI agents faster, cheaper, and more efficient.





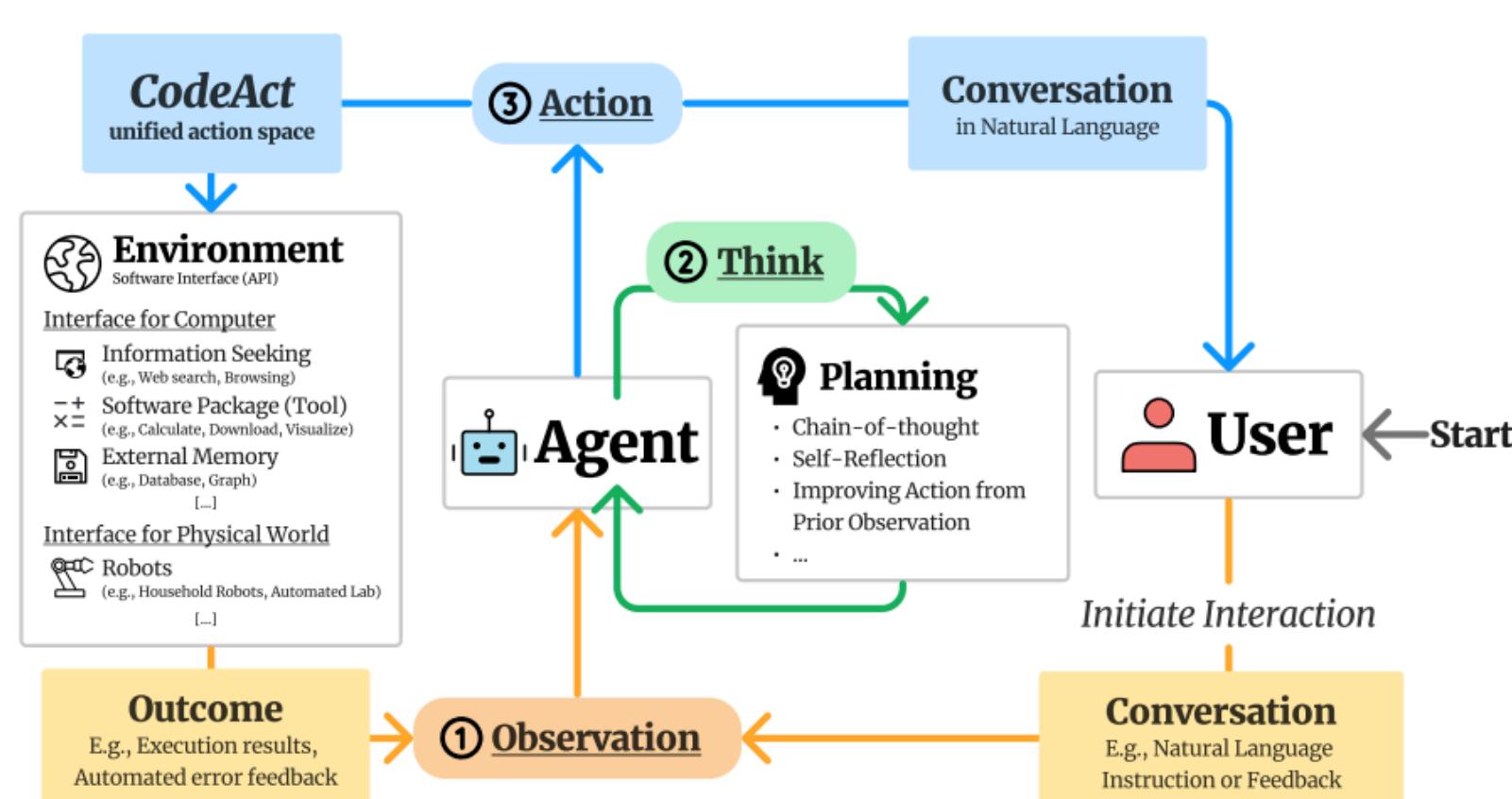
CodeAct - Reason, Code, Execute, Observe

CodeAct uses executable Python code to perform complex, multi-step tasks, avoiding the limits of predefined functions or JSON formats. Traditional LLM agents are restricted by fixed action spaces, but CodeAct provides a unified and flexible action space.

By integrating with a Python interpreter, it can execute actions, revise previous steps, and adapt dynamically through multi-turn interactions.

Manus AI adopted this principle to build their agent with high precision.

Multi-agent setup (planner, executor, knowledge, verifier) orchestrated in an iterative loop: analyze → plan → execute → observe → refine



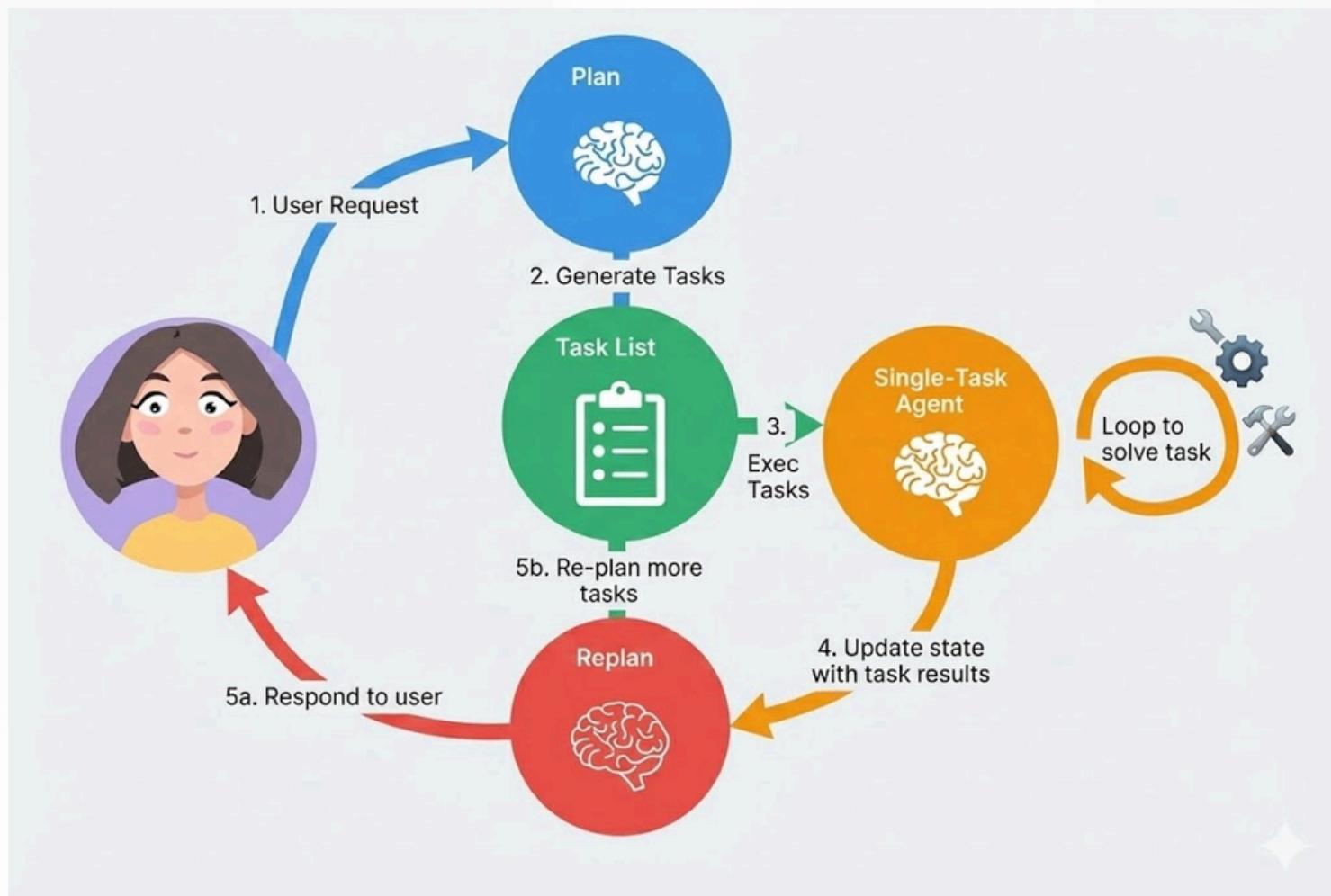


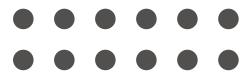
Plan & Execute

Unlike ReAct—where the LLM must think, act, and observe at every step—plan-and-execute agents generate a full multi-step plan up front, allowing sub-tasks to run independently. This reduces repeated LLM calls, lowers costs, and enables faster execution while improving task quality through explicit, structured reasoning.

This design pattern is inspired by the "Plan-and-Solve" prompting paper and projects like **BabyAGI**.

ReWOO fixes a complete plan up front and then just fills in evidence, while **plan-and-execute** can re-plan as it goes, adapting the plan based on intermediate results.





**Follow me to stay updated on
Agents, LLM & RAG!**

Stay tuned for Part 2 & Code!

