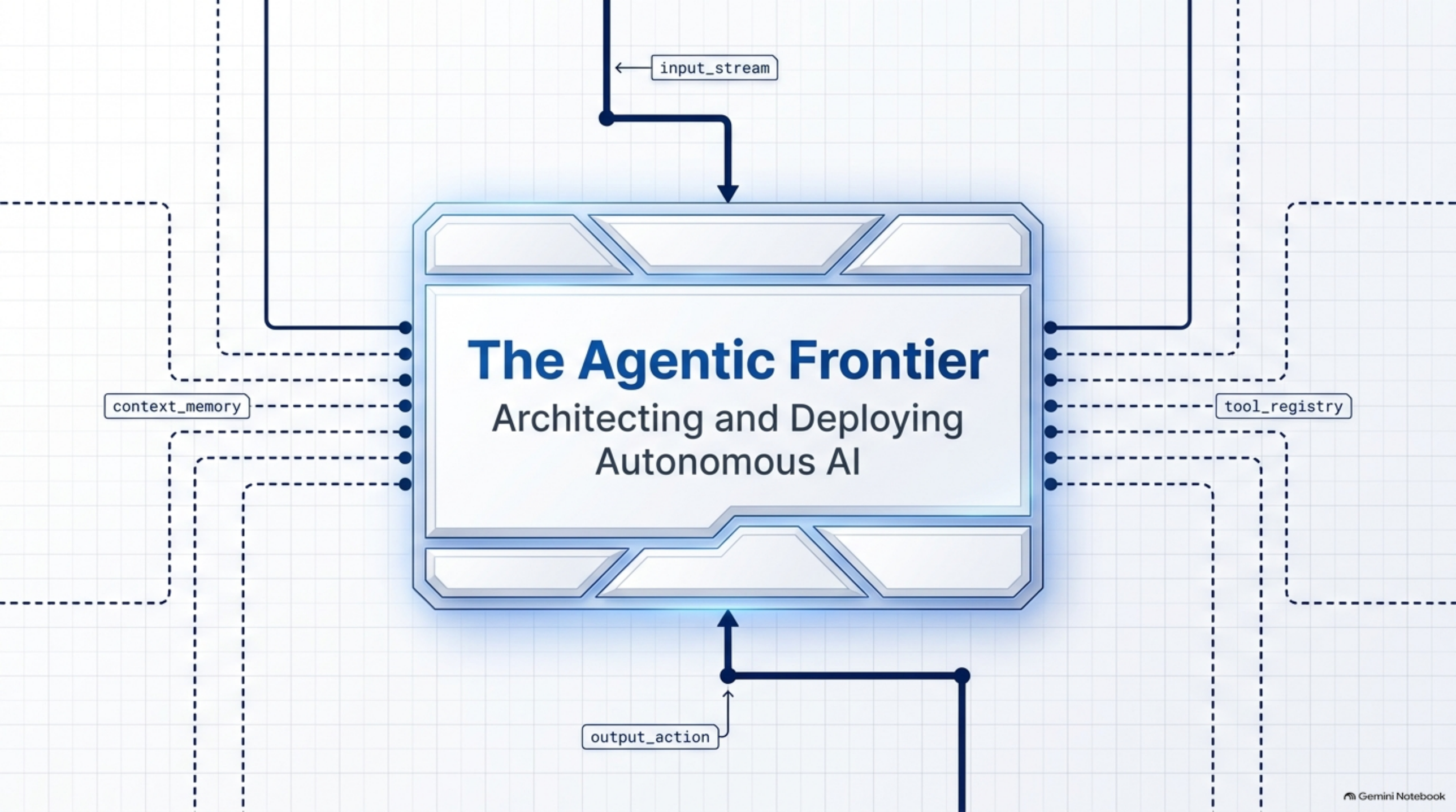
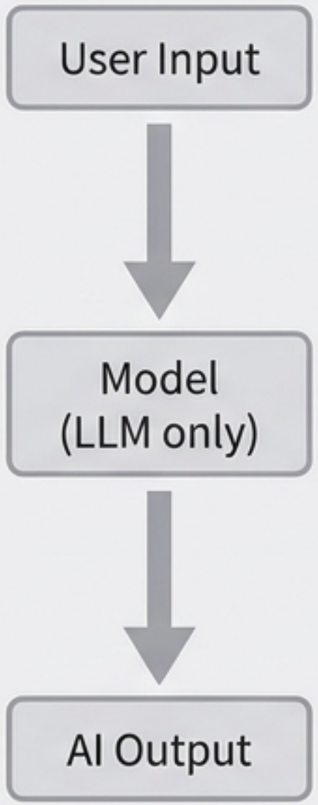


The Agentic Frontier

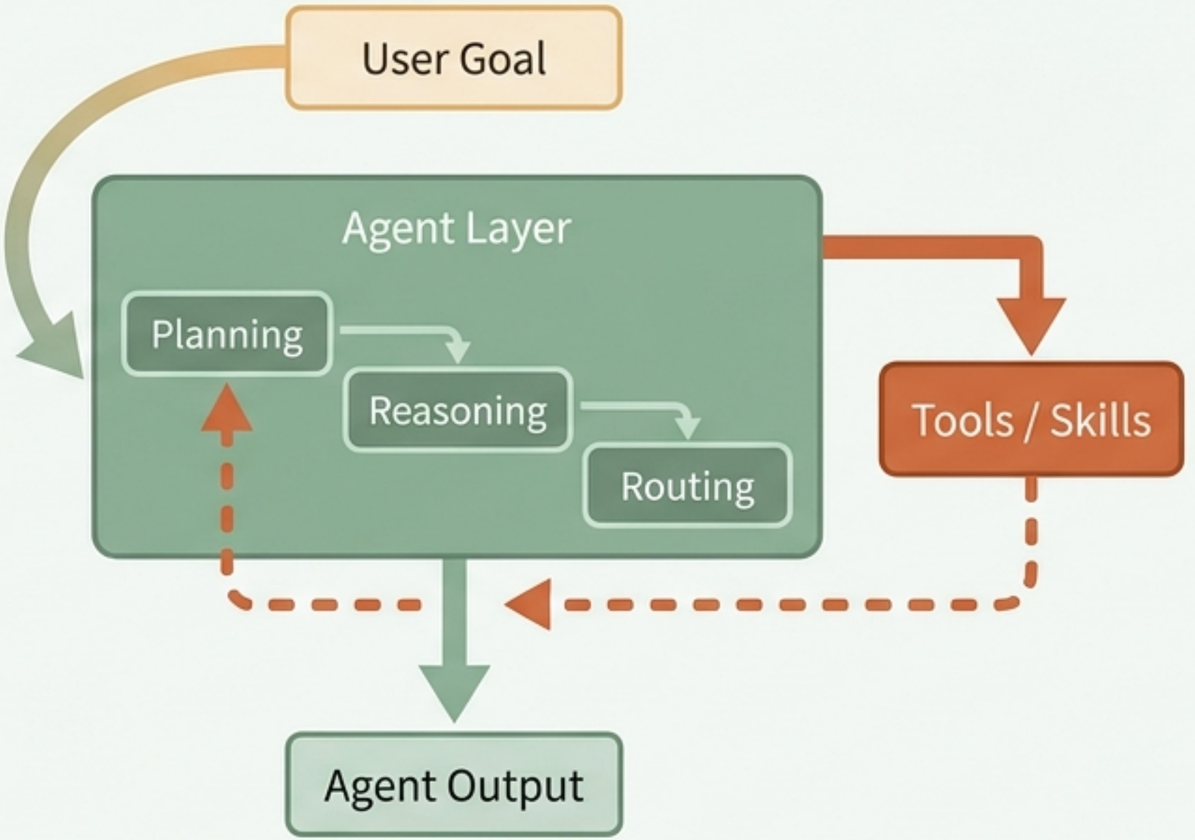
Architecting and Deploying Autonomous AI

NON-AGENTIC AI (Reactive)



Only responds to prompts. Stateless, pure text generation. No planning, no tools.

AGENTIC AI (Autonomous)

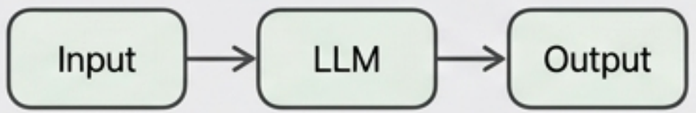


AI with initiative. Decides what to do next, plans multi-step tasks, calls functions, and maintains state.

The AI Capability Spectrum

Single LLM

`Stateless`

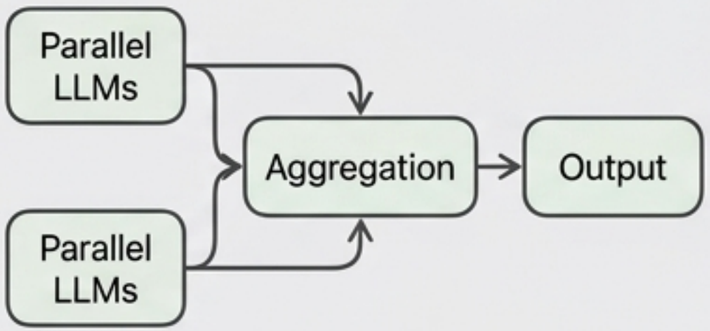


Fast, low cost.
Not adaptable, lacks context.

Summarization, classification, translation.

Structured Workflow

`Deterministic`

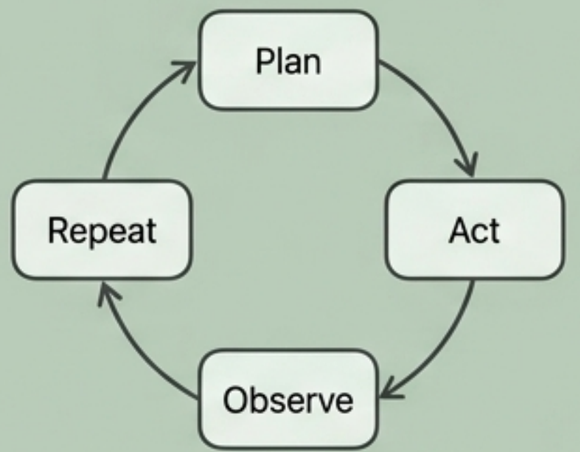


Predictable, easy to audit.
Rigid, not dynamic.

Repetitive, multi-step tasks, compliance-heavy logic.

Autonomous Agent

`Dynamic Planning`

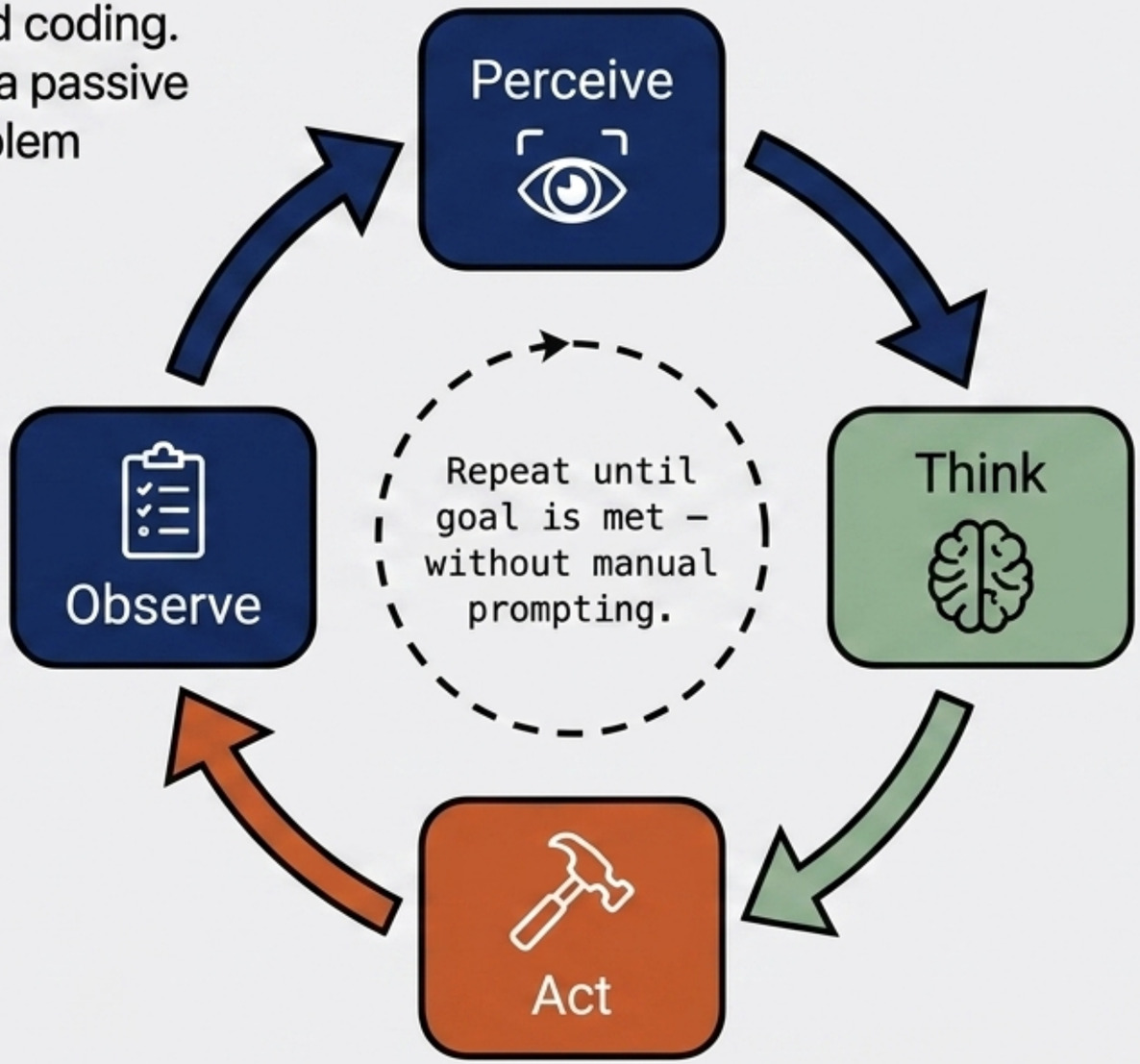


Highly adaptable, learns from feedback.
Unpredictable, higher complexity.

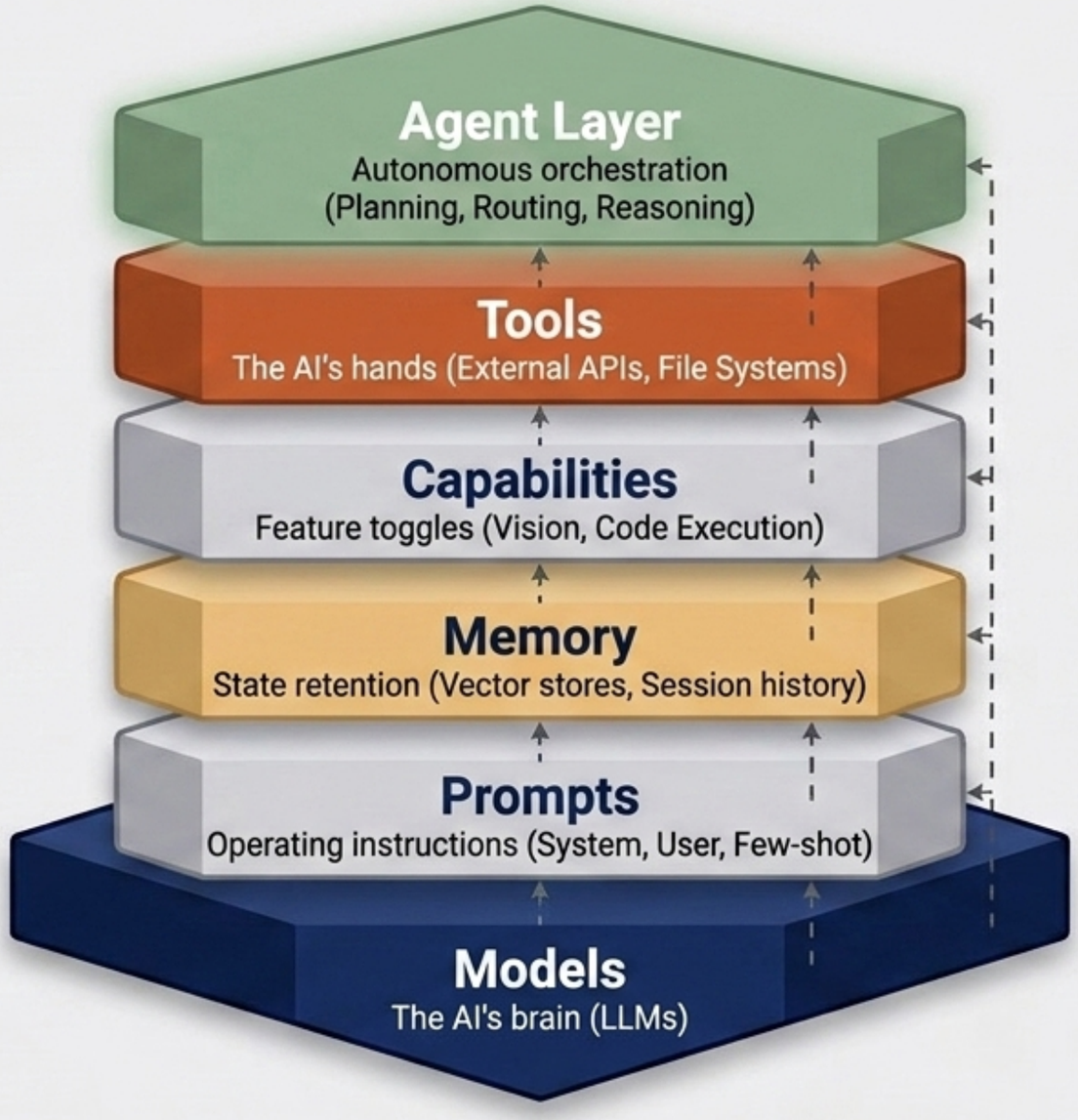
Complex, open-ended tasks with unclear solution paths.

Anatomy of an Agent: The ReAct Loop

Dynamic planning over rigid coding. The agent transitions from a passive responder to an active problem solver.



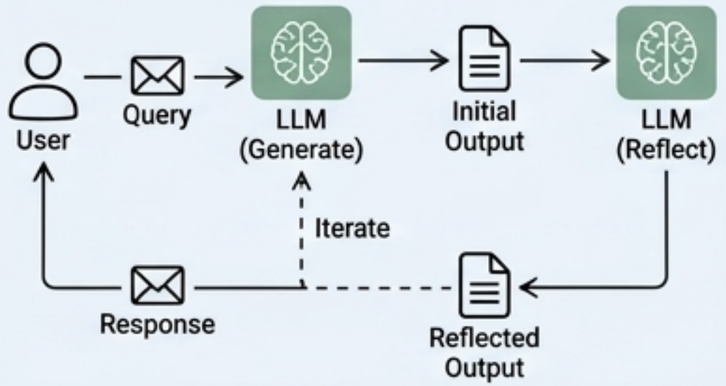
The Agentic Stack: Capabilities & Components



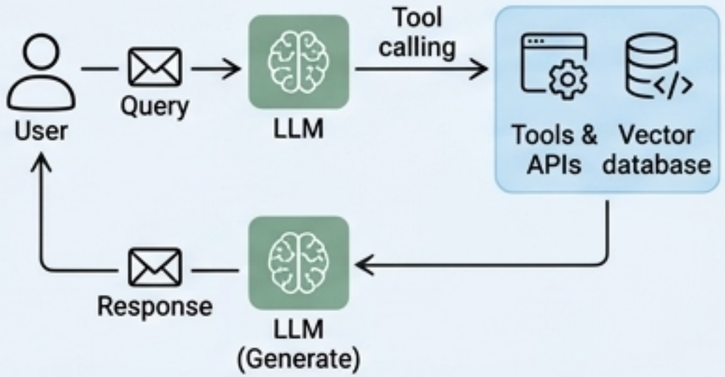
The Action Layer: How Agents Interact with the World

Designing the Brain: Top 5 Agentic Patterns

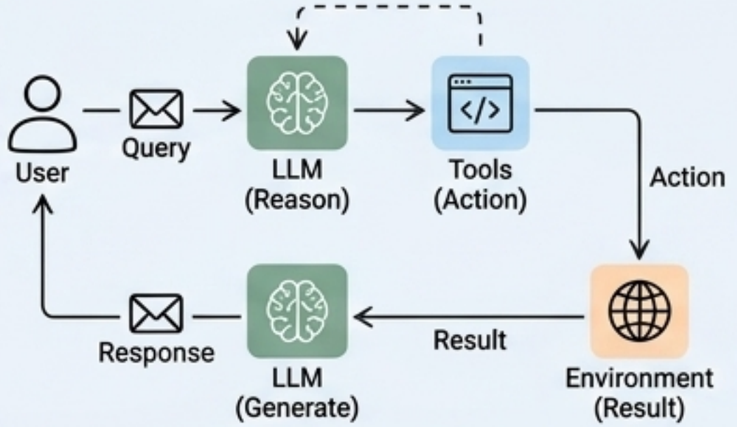
1) Reflection Pattern



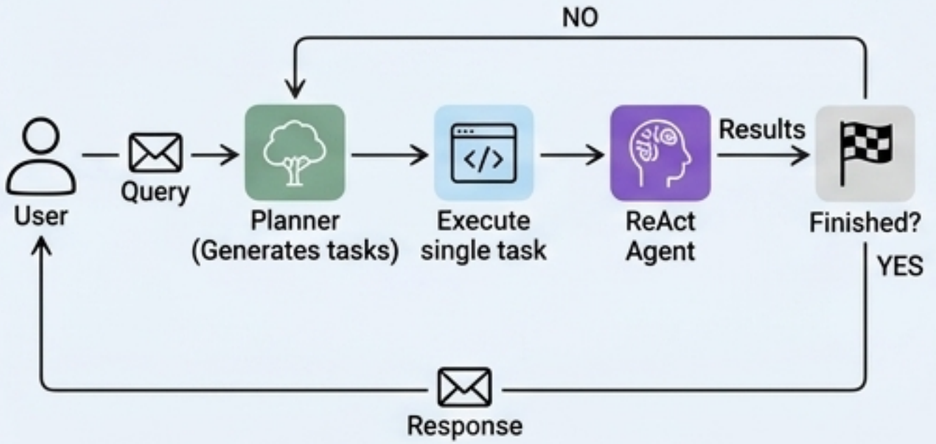
2) Tool Use Pattern



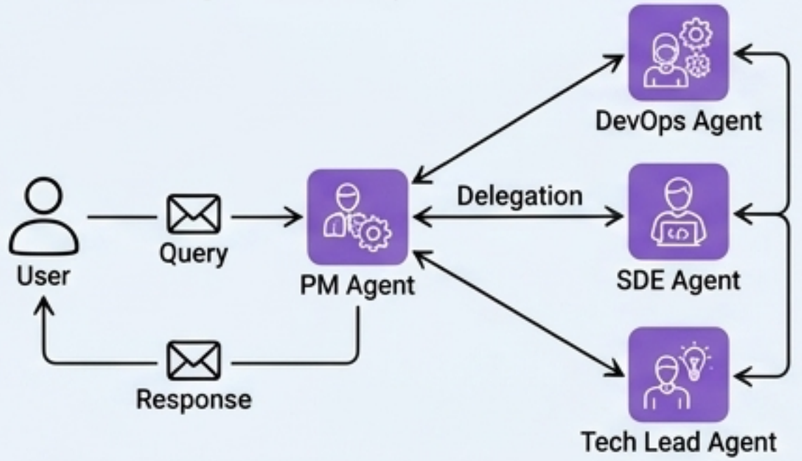
3) ReAct Pattern



4) Planning Pattern

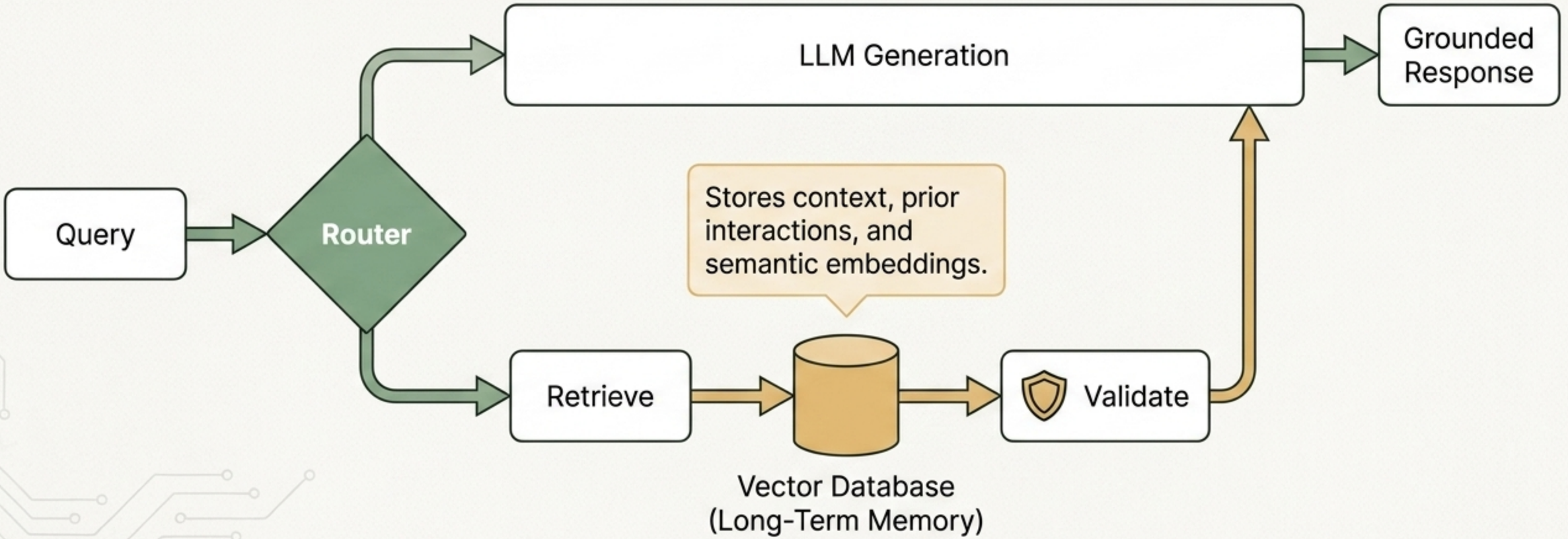


5) Multi-agent Pattern



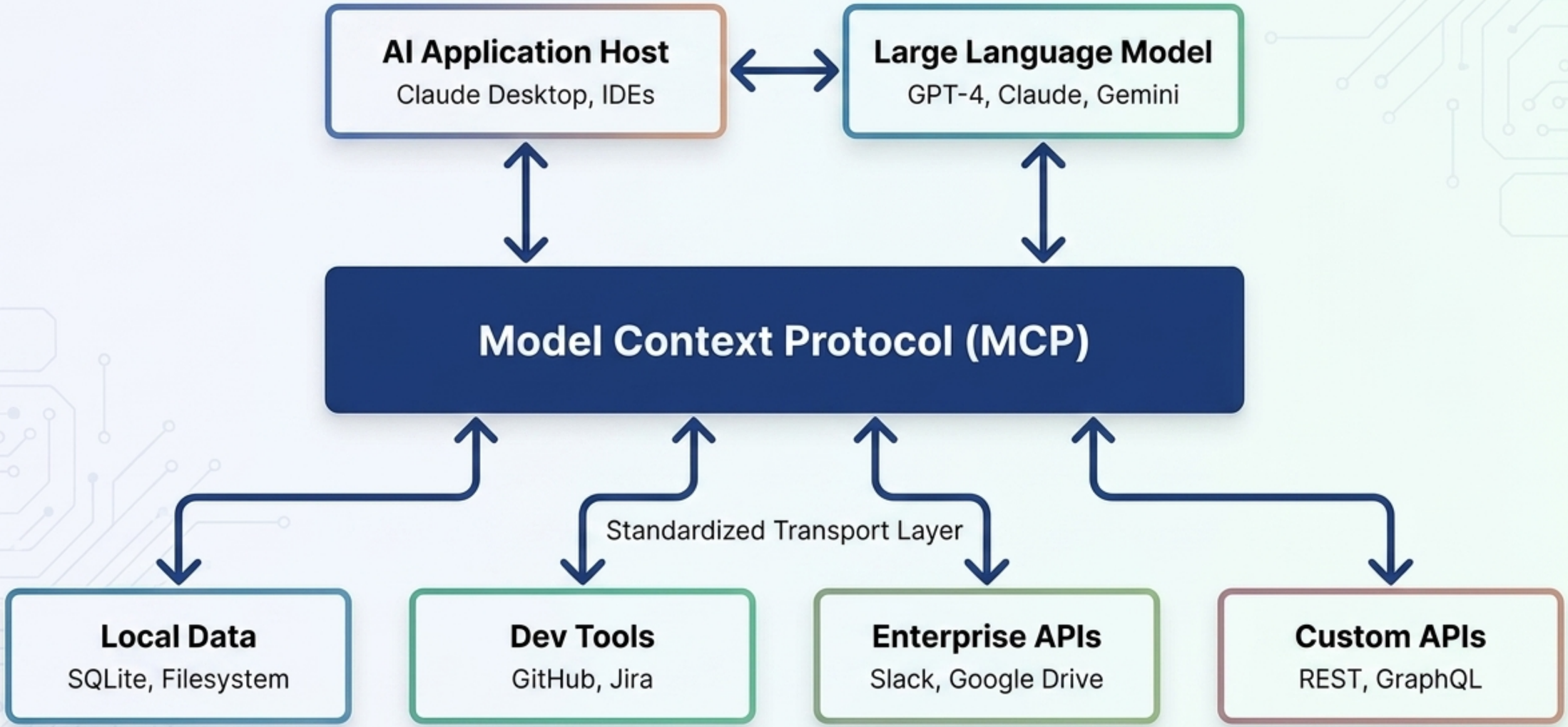
Memory & Context: Agentic RAG

Dynamically routing queries, retrieving embeddings, and validating information to eliminate hallucinations.



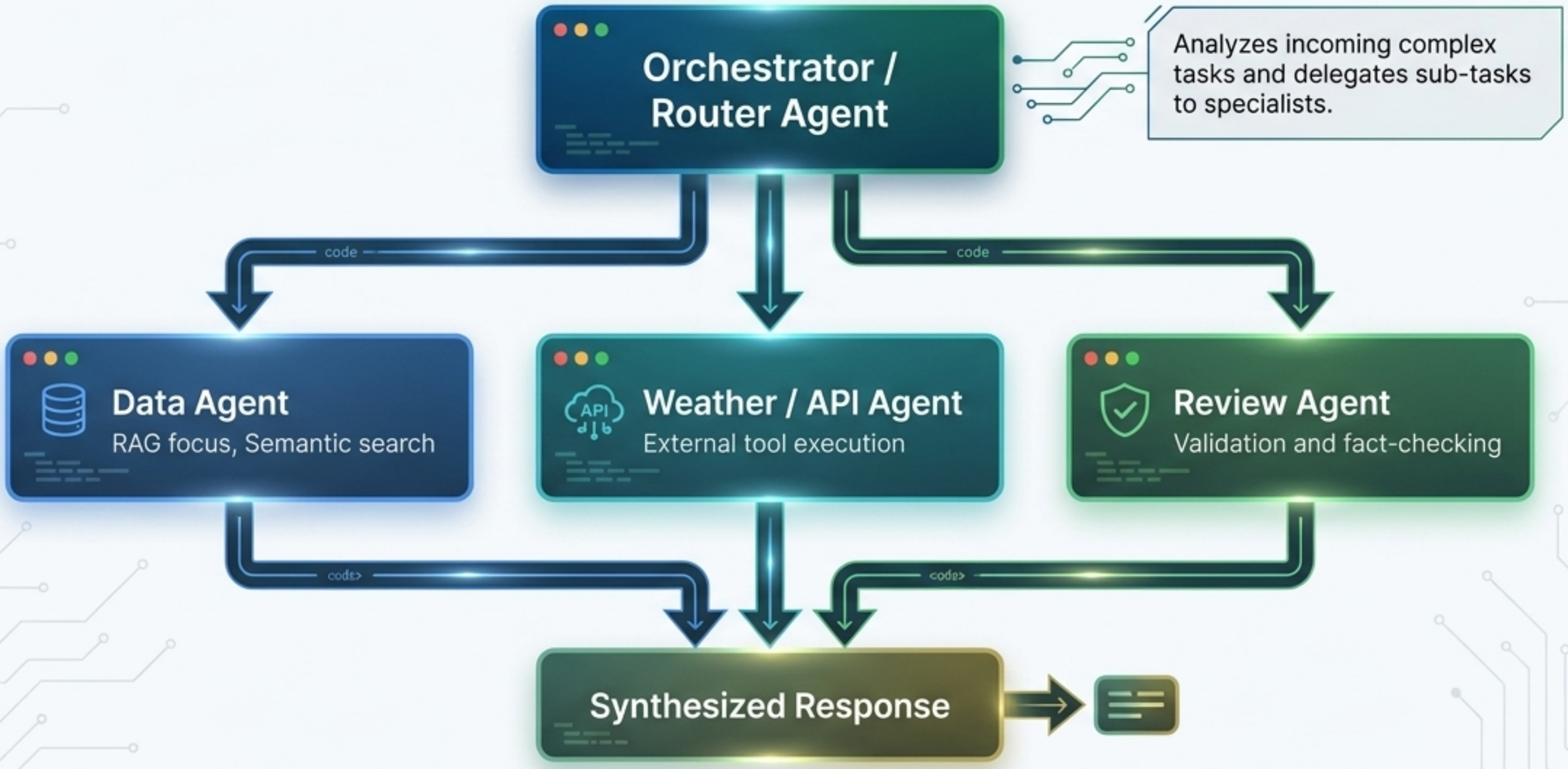
The Universal Standard: Model Context Protocol (MCP)

MCP acts as the USB-C for AI—replacing custom integrations with one secure, universal protocol.



Divide & Conquer: Multi-Agent Architecture

Trading simplicity for scale. Complex tasks are decomposed and distributed to highly accurate, specialized agents.



Build Strategy: From Scratch vs. Frameworks

	Building From Scratch Manual implementation	**Using Frameworks** LangChain, CrewAI
Control	**Total**. Manual API calls and explicit memory management. ✅	**Abstracted**. Framework handles input processing and orchestration. ⚠️
Complexity	**High**. Requires custom boilerplate for routing and asynchronous processes. ⚠️	**Low**. Pre-built memory, routing, and tool-calling mechanisms. ✅
Flexibility	**Unbounded**. Implement bespoke logic without constraints. ✅	**Bounded** by the framework's design patterns. ⚠️
Best For	Learning deep architecture; highly bespoke constraints. ✅	Rapid development; scaling complex multi-agent systems quickly. ✅

Structuring the System Prompt: The BRIDGE Framework

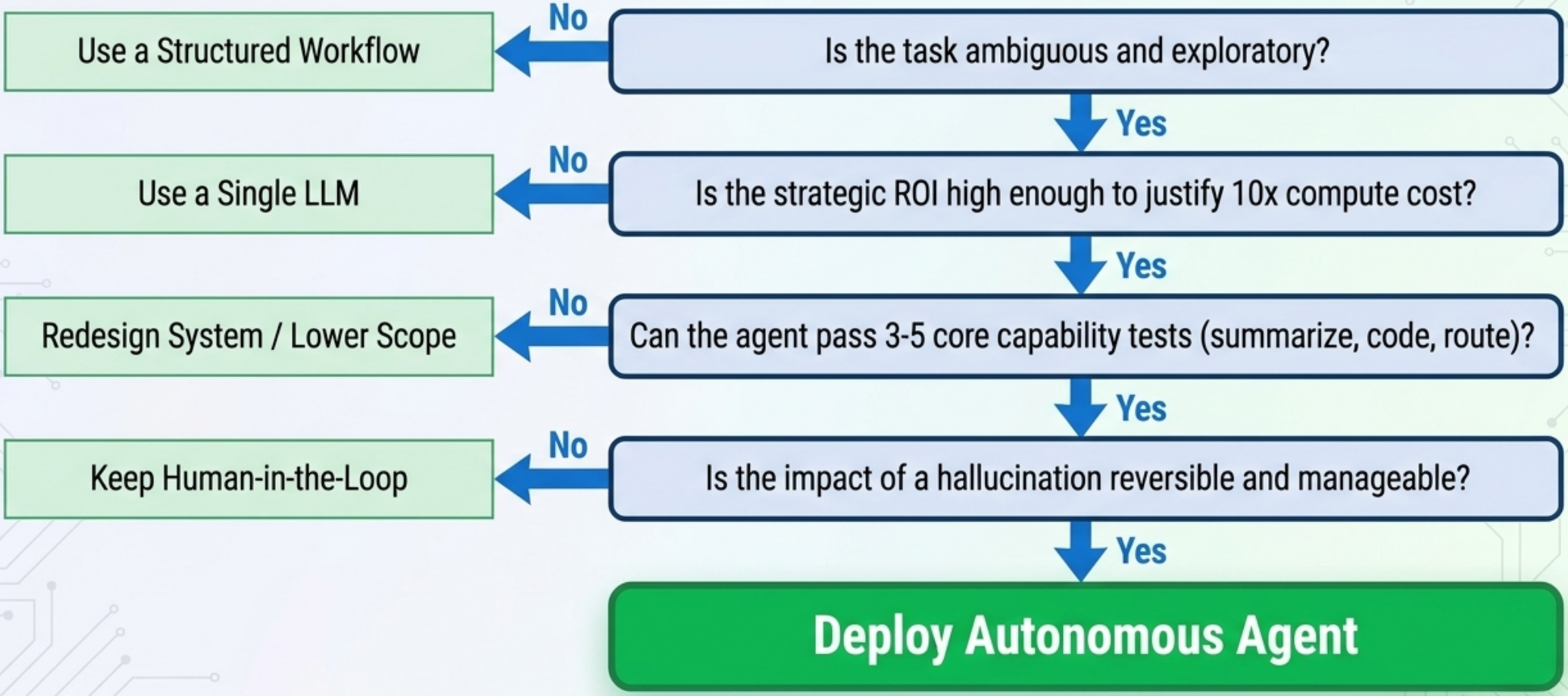
Explicitly separate context from guardrails to construct robust, deterministic operating instructions.

- **B** - **Background:** Provide the business context. Helps the AI understand the "why".
- **R** - **Role:** Define the precise persona (e.g., "Act as a senior DevOps engineer").
- **I** - **Instructions:** State the core multi-step request or primary objective.
- **D** - **Details:** Specify formatting, target audience, and structural constraints.
- **G** - **Guidance:** Outline guardrails (what to prioritize, what to explicitly avoid).
- **E** - **Examples:** Provide few-shot examples to evaluate and anchor the desired output.

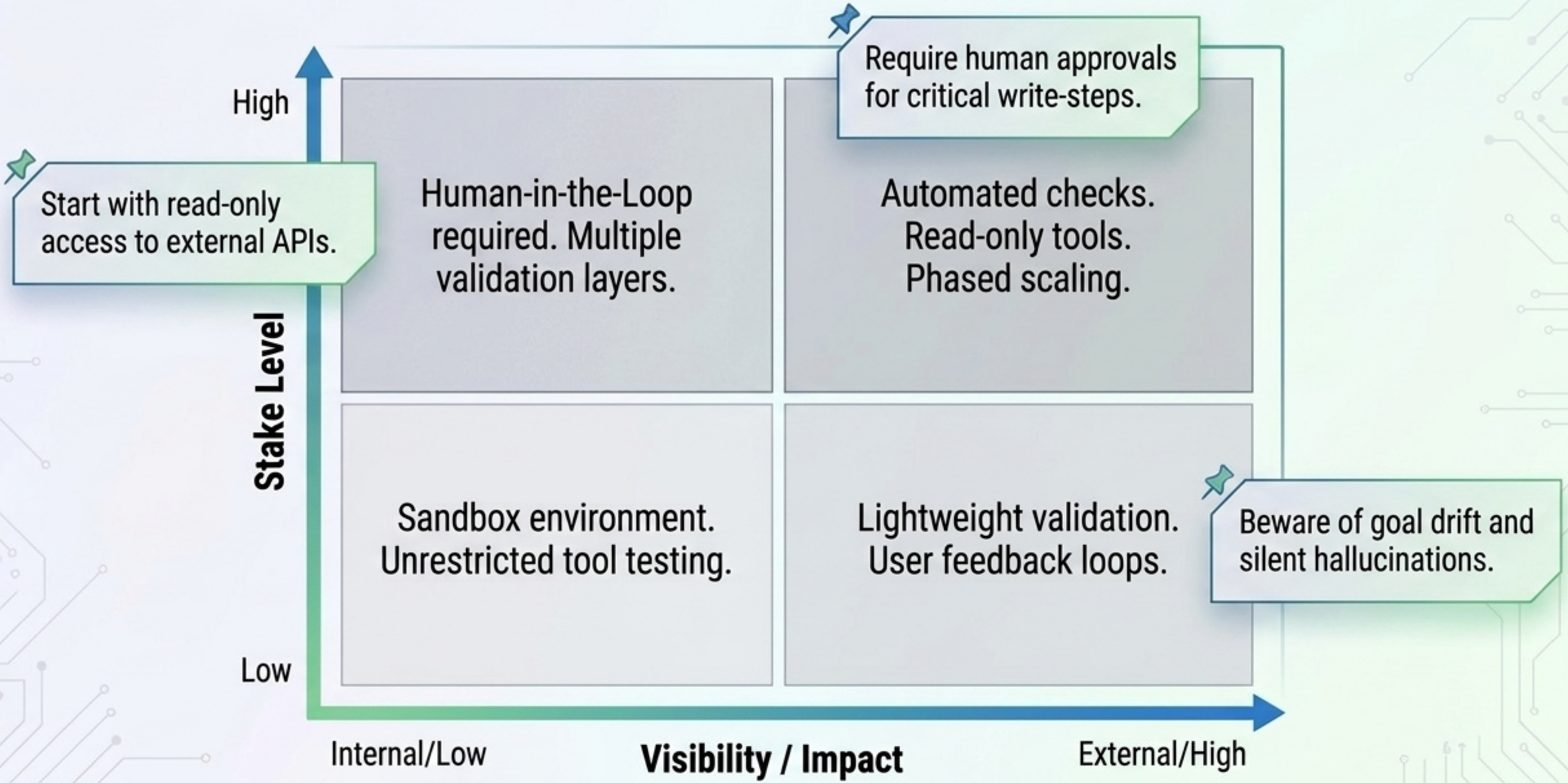
> Prompt engineering is not about typing natural language; it is about **programming deterministic parameters**.

The Decision Matrix: When to Actually Deploy an Agent

Avoid agents for zero-error systems or simple, high-volume tasks.



Risk Management & Deployment Strategy



The Fully Automated Agentic Workflow

Integrating memory, specialized agents, and dynamic tools to transform raw inputs into autonomous synthesis.