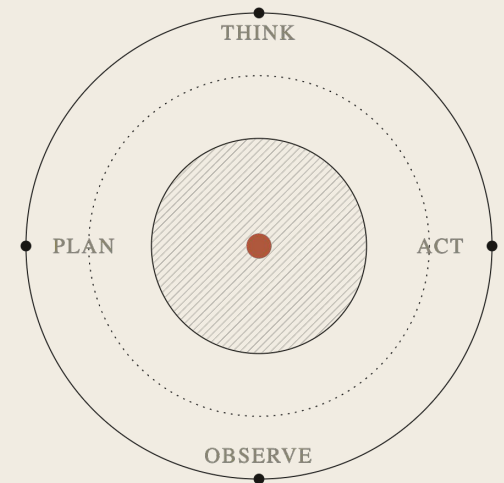


Introduction to *Agents.*

The natural evolution of language models — made useful in software.



WHAT IT IS

A complete application that plans, acts, and observes to reach a goal.

WHY NOW

From passive prediction to autonomous problem-solving and task execution.

IN THIS DECK

Anatomy, the operational loop, a taxonomy, and the path to production.

Four components, *working in a loop.*

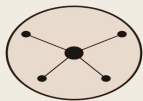
An agent is the model, the tools, an orchestration layer that runs the loop, and the runtime that makes it a service.

01

the brain

Model

The reasoning core. Processes context, evaluates options, decides the next step. Choose for the task, not the benchmark.



02

the hands

Tools

The bridge to the world. APIs, code, databases, retrieval. They let the agent look things up and change things.



03

the nervous system

Orchestration

Runs the Think–Act–Observe loop. Manages memory, planning strategy, and which tool to invoke when.



04

the body

Runtime

Hosting, sessions, persistence, monitoring. Turns a prototype into a service that other people and agents can reach.



The anatomy, in

A minimal agent in Google's ADK — model, instruction, and tools wired into a runner.

```
agent.py

# Define the agent root_agent = Agent (

name="helpful_assistant" ,

model=Gemini(model="gemini-2.5-flash-lite" ),

description="A simple agent for general Q&A." ,

instruction="You are a helpful assistant. Use "Google Search for current info." ,

tools =[google_search],
)

# Host it, then send a mission runner = InMemoryRunner(agent =root_agent)

response =

await runner.run_debug (

"What is Agent Development Kit from Google?" )
```

THE BRAIN

Model

A Gemini variant chosen for the task — fast and cheap for general Q&A.

THE CONSTITUTION

Instruction

The system prompt — persona, scope, and tool-use guidance.

THE HANDS

Tools

Capabilities the agent can call — here, `google_search` for live grounding.

THE BODY

Runner

Hosts the agent, runs the loop, dispatches the mission.

The minimum viable agent stack,
2026.

Default tools at each layer — when to move past them, and
what shifted since 2024.

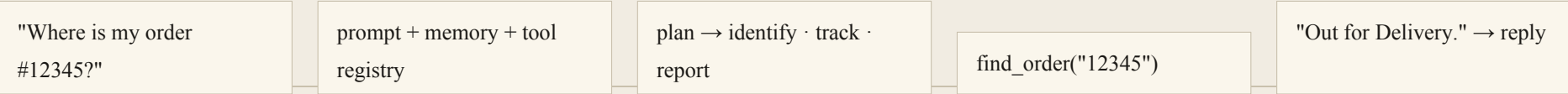
LAYER	START HERE	WHEN TO UPGRADE	WHAT CHANGED SINCE 2024
Model Serving	Anthropic or OpenAI API	Self-host when cost or latency demands it	Reasoning models. Open-weight gap closing.
Protocols & Tools	MCP	You don't. It's the standard.	Didn't exist. Now ~2,000 servers, Linux Foundation.
Memory & Knowledge	Postgres (pgvector) + in-context memory	Dedicated vector DB at 10M+ embeddings	Memory elevated from subset of storage to own layer.
Frameworks	Provider SDK (simple) or LangGraph (complex)	When you outgrow the SDK	Provider SDKs entered. Framework vs. SDK is the real question.
Eval & Observability	Langfuse or Braintrust	When you need custom evals at scale	Layer didn't exist in 2024.
Guardrails & Safety	NeMo Guardrails + your own policy layer	When agent count exceeds manual review	Agent guardrails ≠ LLM guardrails. New discipline.

Sources — LangChain State of Agent Engineering (Dec 2025); Anthropic MCP (Dec 2025); LangGraph 1.0 (Oct 2025).

Mission to result, in *five* steps.

A continuous cycle of perception, reasoning, action, and observation — until the goal is met.

01	02	03	04	05
RECEIVE	PERCEIVE	REASON	EXECUTE	LOOP
Get the mission	Scan the scene	Think it through	Take action	Observe & iterate
A user request or an automated trigger sets a specific, high-level goal.	Gather context — the prompt, memory, available tools and data.	The model devises a multi-step plan from the goal and the context.	Invoke a tool — call an API, run code, query a database.	Add the result to context and return to step three — until the goal is met.



Five levels of *agentic capability*.

Scope this first. The level you choose decides the architecture, the tooling, and the operational lift.

L 0	Core Reasoning	A language model alone. Strong on what it was trained on, blind to anything new.	■□□□□
L 1	Connected Problem-Solver	Adds tools. Can search, retrieve, query — grounded in real, current information.	■ ■ □ □ □
L 2	Strategic Problem-Solver	Plans multi-step work, curates its own context, and chains tools toward a complex goal.	■ ■ ■ □ □
L 3	Collaborative Multi-Agent	A team of specialists. A coordinator delegates sub-tasks to agents that treat each other as tools.	■ ■ ■ ■ □
L 4	Self-Evolving System	Identifies its own capability gaps and creates new tools — or new agents — to fill them.	■ ■ ■ ■ ■

A prototype is easy. *Production is the work.*

Stochastic systems need a new operational discipline — measurement, judgement, and traces.

MEASURE

Instrument like an experiment

Define KPIs that map to business outcomes — goal completion, satisfaction, latency, cost per interaction.

Goal completion rate

Task latency & cost

Revenue / retention impact

JUDGE

Quality over pass / fail

A pass/fail unit test cannot grade prose. Use an LM-as-Judge against a curated golden dataset to score quality.

Golden eval dataset

Rubric-based scoring

Domain-expert review

TRACE

Debug the trajectory

OpenTelemetry traces record every prompt, tool call, parameter, and observation — the only way to answer *why*.

Full reasoning trajectory

Tool I/O capture

Human feedback loop

Building an agent is closer to directing than bricklaying — you set the scene, cast the tools, and supply the context. The agent does the rest.