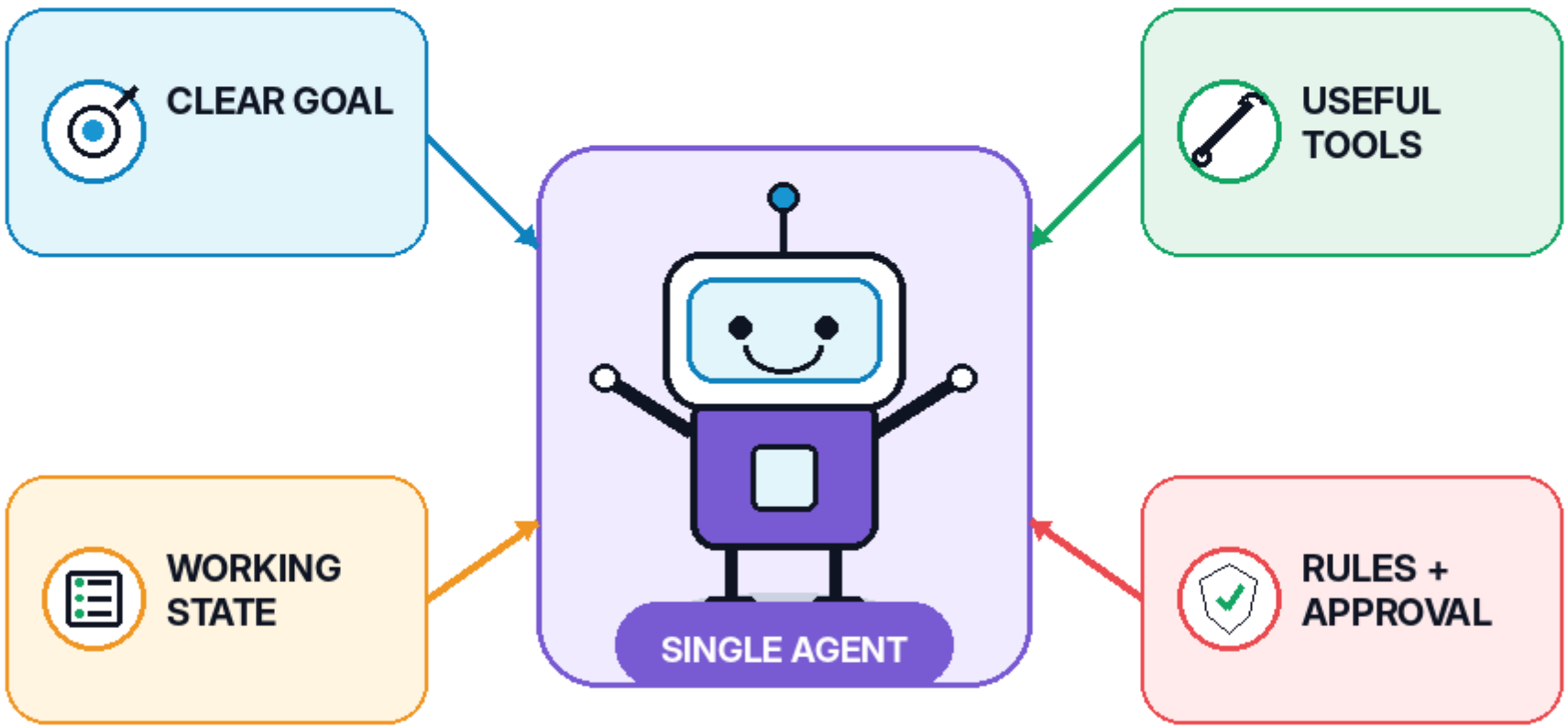
AGENTIC AI SERIES: DAY 11

BUILDING A CAPABLE SINGLE AGENT

One agent. One clear job. Done reliably.



Before building an AI team, build one agent that can finish a narrow job.

AGENTIC AI SERIES: DAY 11

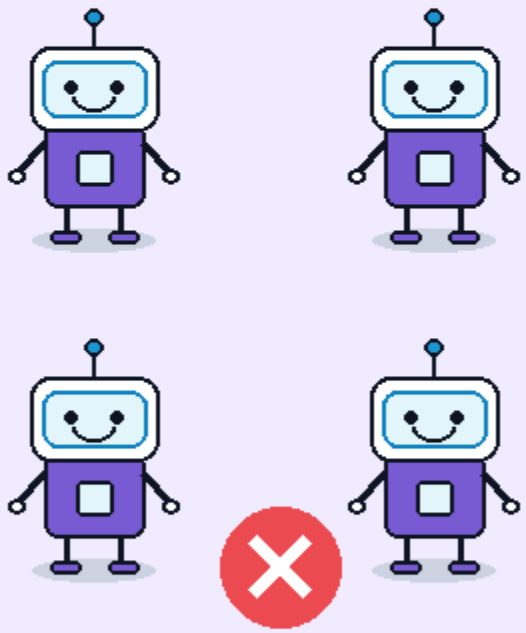
ONE SKILLED WORKER BEFORE A WHOLE TEAM

A simple analogy: one restaurant table does not need five managers.

ONE CAPABLE MANAGER

- Take order
- Check kitchen
- Update customer
- Handle small issue

UNNECESSARY TEAM



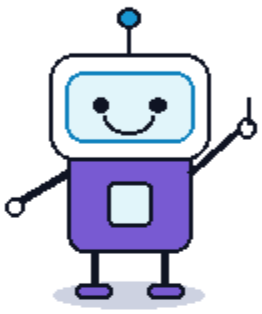
More coordination • more cost • more failure points

Build one agent well. Add more agents only when the work truly needs them.

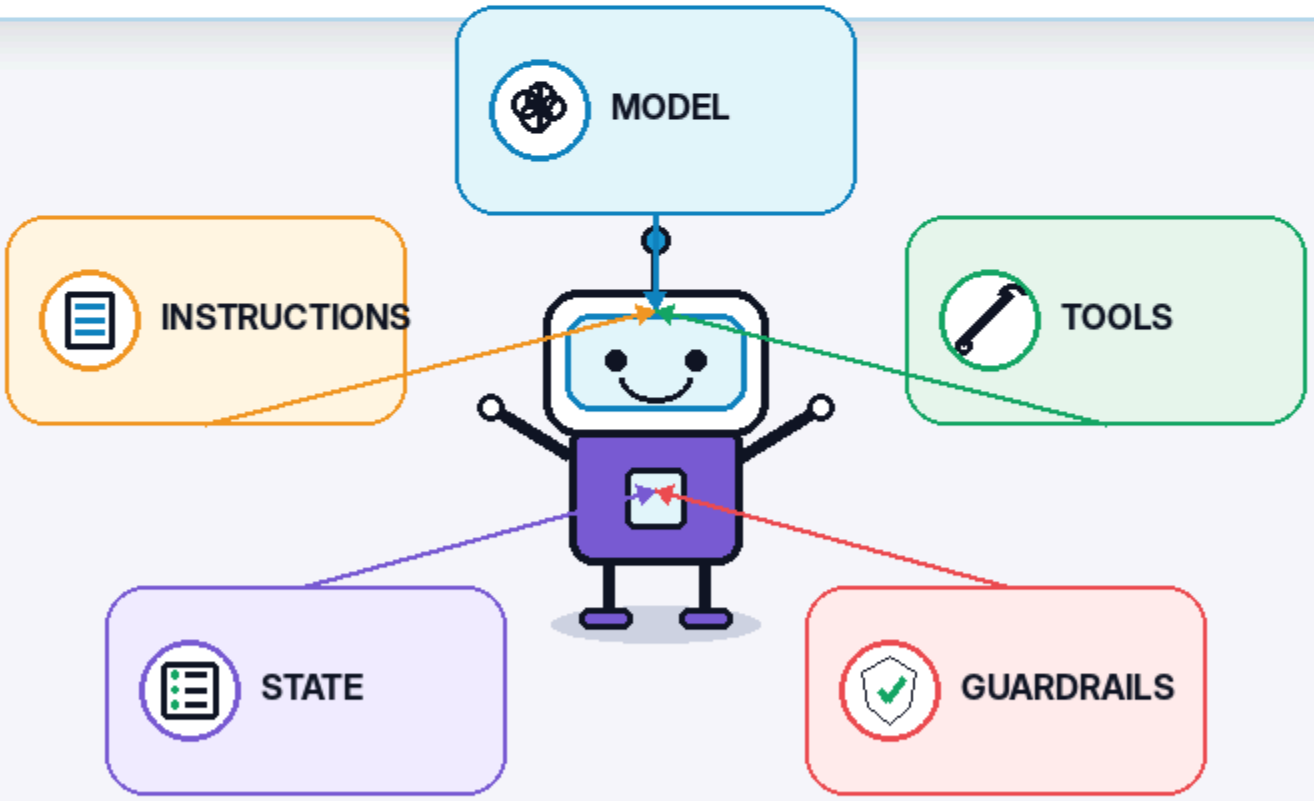
AGENTIC AI SERIES: DAY 11

WHAT IS A SINGLE AGENT?

One decision-maker can use several tools many times and still remain one agent.



A single agent is one AI decision-maker responsible for one goal. It understands the request, chooses actions, uses tools, updates its state, and stops or asks for help.



One agent does not mean one tool call. It means one central decision-maker.

AGENTIC AI SERIES: DAY 11

PICK ONE JOB—
NOT AN ENTIRE COMPANY

A capable agent starts with a narrow, understandable mission.

TOO BROAD

"Handle all customer support."

Too many products, policies, actions, risks, and exceptions.

CLEAR SCOPE

"Track delivery status and resolve standard refund requests for recent orders."

Clear users, data, tools, limits, and handoff rules.

A narrow job is easier to teach, test, secure, and improve.

AGENTIC AI SERIES: DAY 11

DEFINE "DONE" BEFORE YOU BUILD

The agent needs a finish line—not just a starting instruction.

SUCCESS CHECKLIST

Correct outcome	The user receives the right answer or action.
Supported by evidence	The result comes from trusted data or tool output.
Within permission	The agent never exceeds the user's access.
Within limits	It respects time, cost, and retry budgets.
Safe handoff	It asks a human when uncertain or blocked.

Example: refund submitted + reference ID returned + customer notified.

If you cannot explain what success looks like, the agent cannot reliably stop.

AGENTIC AI SERIES: DAY 11

THE 6 PARTS OF A CAPABLE SINGLE AGENT

Think of the agent as a complete system—not just a model.

1. MODEL

The brain that reasons and selects actions.

2. INSTRUCTIONS

The job description, rules, and stop conditions.

3. TOOLS

The hands that search, calculate, read, or act.

4. CONTEXT

The relevant information needed right now.

5. STATE

The task checklist: what happened and what remains.

6. GUARDRAILS

Permissions, validation, approvals, and limits.

AGENTIC AI SERIES: DAY 11

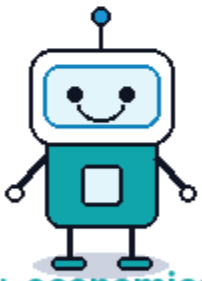
MODEL = THE BRAIN, NOT THE WHOLE AGENT

Choose enough intelligence for the job—then add the surrounding system.

SIMPLE TASK

Example:

"Look up this order and explain its status."



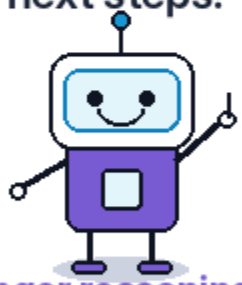
Fast + economical

MODEL / LLM

HARDER TASK

Example:

"Compare several policies, resolve conflicts, and plan next steps."



Stronger reasoning

A larger model cannot fix vague goals, poor tools, missing data, or unsafe permissions.

AGENTIC AI SERIES: DAY 11

INSTRUCTIONS = THE JOB DESCRIPTION

Tell the agent who it is, what it owns, how it works, and where it must stop.

SUPPORT AGENT BRIEF

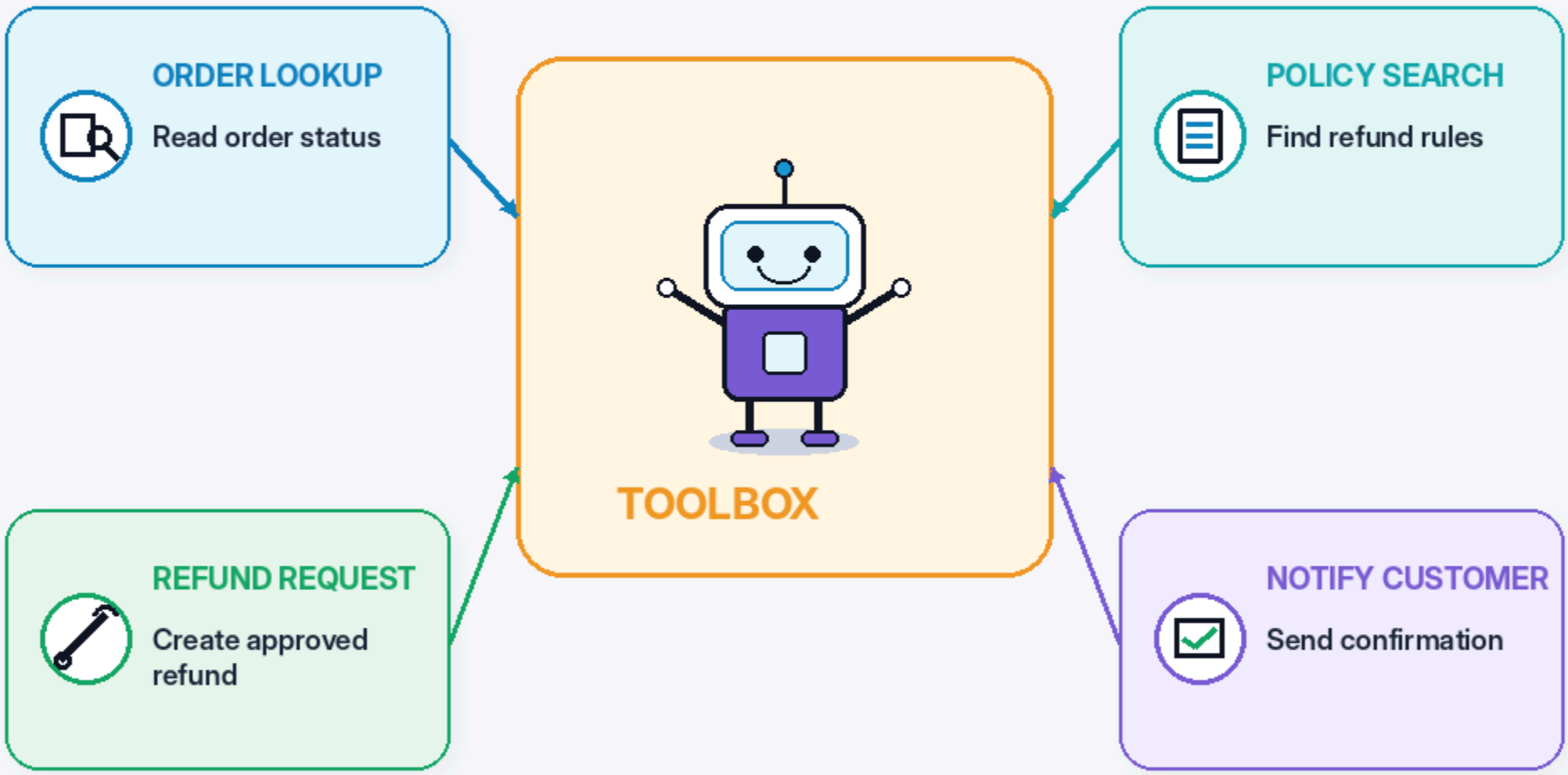
ROLE	You help customers with delivery status and standard refunds.
GOAL	Resolve eligible cases accurately and quickly.
PROCESS	Verify identity → inspect order → check policy → act or hand off.
BOUNDARIES	Never change bank details. Never promise unsupported outcomes.
STOP	Finish when resolved, blocked, or human approval is required.
OUTPUT	State what happened, evidence used, action taken, and next step.

Good instructions are specific and testable—not merely long.

AGENTIC AI SERIES: DAY 11

TOOLS = THE AGENT'S HANDS

The model can decide—but tools let the system read live data and take action.



Give the agent only the tools required for its job—and separate read tools from action tools.

AGENTIC AI SERIES: DAY 11

TOOL QUALITY BEATS TOOL QUANTITY

A confusing toolbox creates confusing decisions.

BAD TOOL DESIGN

```
do_everything(request)
```

Vague name

Huge input

Unclear side effects

Unhelpful errors

CLEAR TOOL DESIGN

```
get_order(order_id)
search_policy(topic)
create_refund(order_id, amount)
send_update(customer_id, message)
```

Specific inputs + predictable outputs

Clear names, descriptions, inputs, outputs, and errors help the model choose correctly.

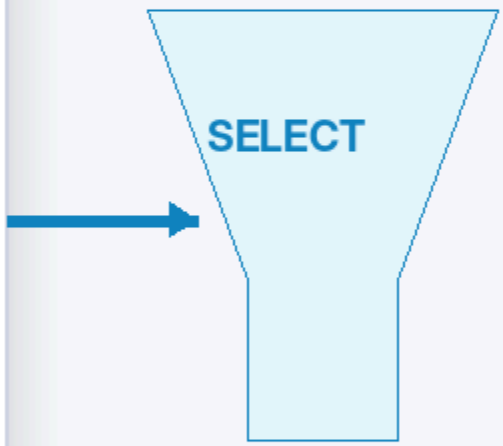
AGENTIC AI SERIES: DAY 11

CONTEXT = WHAT THE AGENT NEEDS RIGHT NOW

More information is not always better. Relevant information is better.

EVERYTHING

- 5 years of chats
- All company policies
- Every customer record
- 40 unused tool results
- Old notes and drafts



RELEVANT

- Current request
- Verified customer + order
- Applicable refund policy
- User permissions
- Latest tool result

Context should be selected for the current decision—not dumped into every step.

AGENTIC AI SERIES: DAY 11

STATE = THE AGENT'S TASK CHECKLIST

State tracks progress so the agent knows what happened, what remains, and when to stop.

CURRENT TASK STATE

GOAL	Resolve damaged-item refund
FACTS FOUND	Order verified · delivered yesterday
COMPLETED	Photo received · policy checked
PENDING	Create refund · send confirmation
RETRY COUNT	1 of 3 attempts used
APPROVAL	Not required: amount below limit

State is not "remember everything." It is the minimum record needed to continue safely.

Without state, agents repeat work, lose evidence, or continue after the task is already complete.

AGENTIC AI SERIES: DAY 11

GUARDRAILS = SEATBELT + BRAKES

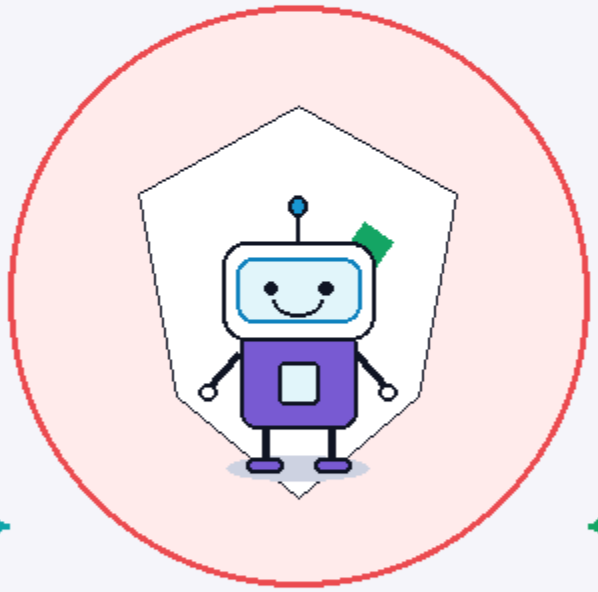
The agent needs freedom to work—and controls that limit the blast radius.

INPUT CHECK

Is the request allowed?

TOOL CHECK

Are arguments and results safe?



PERMISSION

Can this user access it?

APPROVAL

Pause before payments, edits, or deletion.

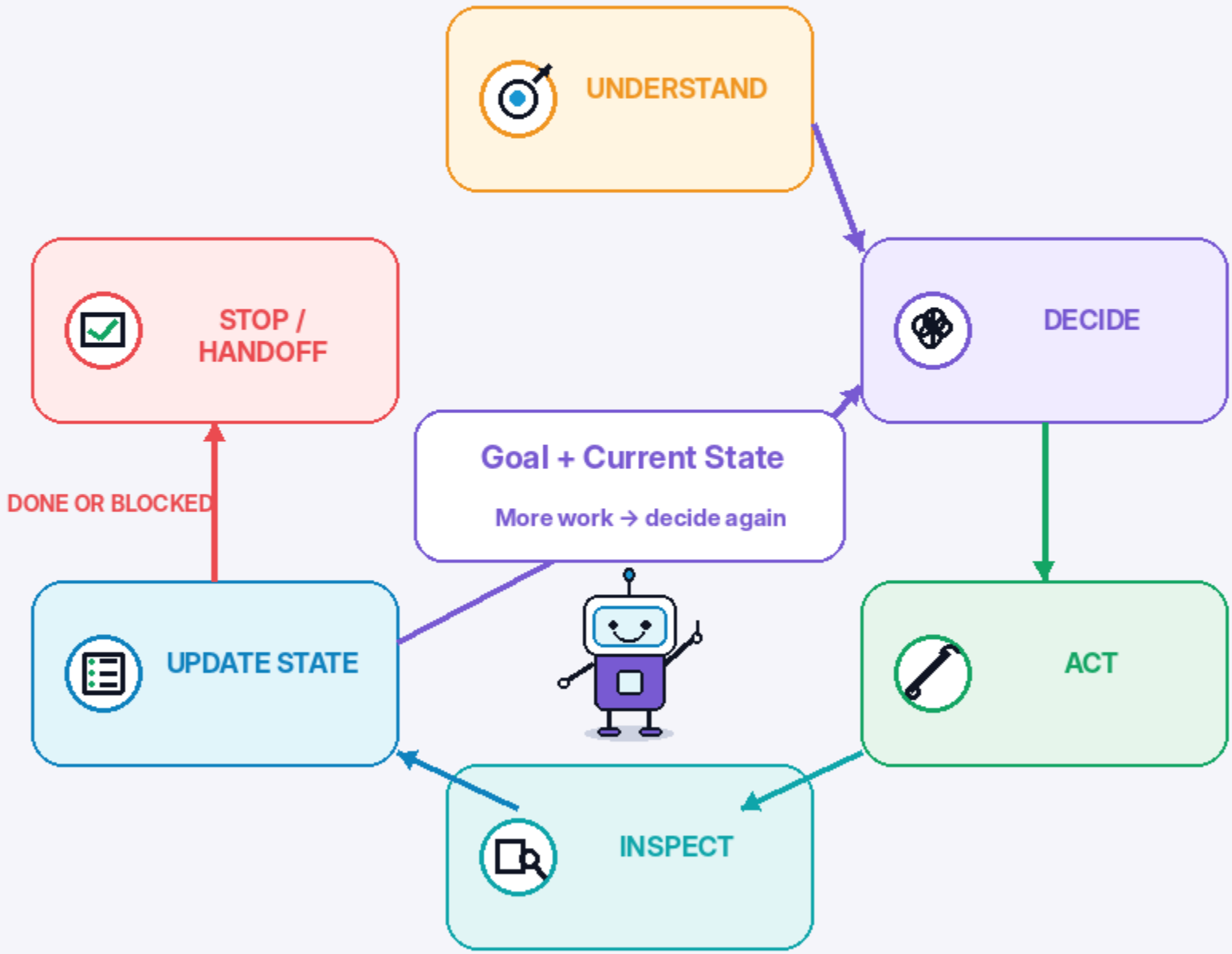
Also set: max steps · timeouts · retry limits · cost budgets · human handoff

A capable agent is not the one with the most freedom—it is the one with the right freedom.

AGENTIC AI SERIES: DAY 11

THE CAPABLE SINGLE-AGENT LOOP

One decision at a time, with state and safety checks around every action.



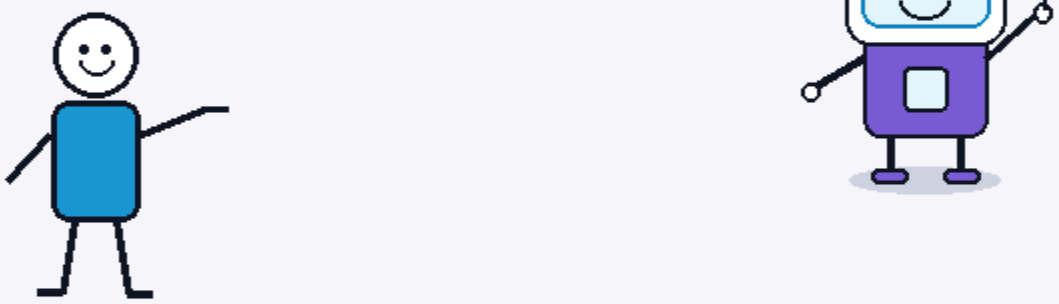
Continue only when the next action is useful, permitted, and within budget.

AGENTIC AI SERIES: DAY 11

FULL EXAMPLE:
DAMAGED DELIVERY REFUND

Watch one agent combine instructions, tools, state, and guardrails.

"My headphones arrived damaged. Can I get a refund?"



1 VERIFY
Check identity and order
`get_order`

2 COLLECT
Ask for damage photo
`user input`

3 CHECK
Read refund policy
`search_policy`

4 DECIDE
Eligible and below approval limit
`state + rules`

5 ACT
Create refund and notify
`refund + message`

Result: refund reference returned, customer notified, and the full action trace recorded.

AGENTIC AI SERIES: DAY 11

SAME ARCHITECTURE, DIFFERENT DOMAINS

The components stay the same. The job, tools, data, and rules change.

STUDY COACH

Goal: build a revision plan

Tools: syllabus + calendar

Rule: do not complete graded work

HR LEAVE ASSISTANT

Goal: answer and submit leave

Tools: policy + HR system

Rule: manager approval when required

IT SUPPORT AGENT

Goal: fix common access issues

Tools: logs + knowledge base

Rule: never reset privileged accounts

SALES REPORT AGENT

Goal: prepare weekly summary

Tools: CRM + calculator + document

Rule: cite numbers and flag missing data

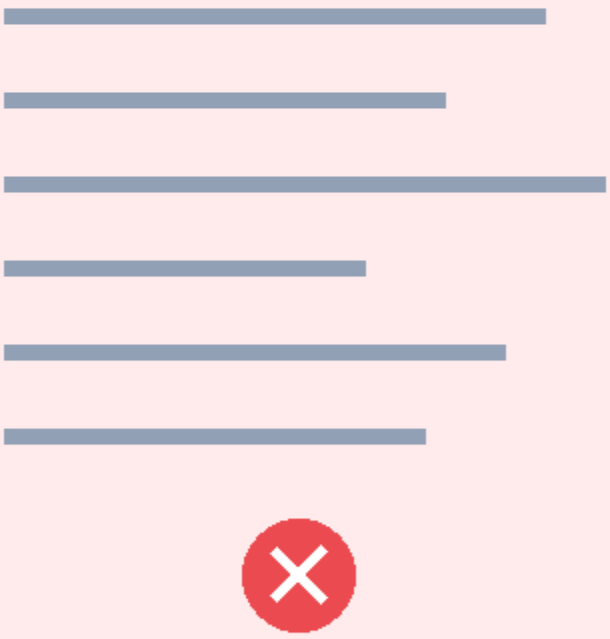
A single-agent pattern can serve many domains when its scope and controls are domain-specific.

AGENTIC AI SERIES: DAY 11

MAKE THE OUTPUT PREDICTABLE

Do not let every completion arrive in a different shape.

FREE-FORM ONLY



STRUCTURED RESULT

STATUS	Refund approved
EVIDENCE	Order #A104 · photo · policy §4.2
ACTION	₹1,499 refund created
NEXT STEP	Customer receives confirmation
HANDOFF	No human review required

Humans see a clear summary. Systems receive fields.

Predictable outputs make agents easier to integrate, validate, audit, and test.

AGENTIC AI SERIES: DAY 11

TEST ONE AGENT
BEFORE ADDING MORE

Complexity should be earned by evidence—not added because "multi-agent" sounds advanced.

NORMAL
Common requests

EDGE
Missing or conflicting data

FAILURE
Tool timeout or bad result

ADVERSARIAL
Prompt injection or misuse

WHAT TO MEASURE

- **Task completed?** Did it reach the real goal?
- **Grounded?** Is the answer supported by evidence?
- **Good handoff?** Did it stop and escalate at the right time?
- **Right tools?** Were calls accurate and necessary?
- **Safe?** Were permissions and approvals respected?
- **Efficient?** Were time, steps, and cost reasonable?

Fix the goal, instructions, tools, context, or guardrails before multiplying agents.

AGENTIC AI SERIES: DAY 11

DAY 11 TAKEAWAY

ONE CLEAR JOB + FEW GOOD TOOLS + USEFUL STATE + GUARDRAILS + TESTS

That is the foundation of a capable single agent.

WHEN ONE AGENT STARTS TO STRUGGLE:

Too many unrelated jobs · too many tools · separate permissions · parallel work · context overload

NEXT UP — Day 12: From One Agent to a Team — Supervisors & Handoffs

What is one narrow job you would give your first capable agent?