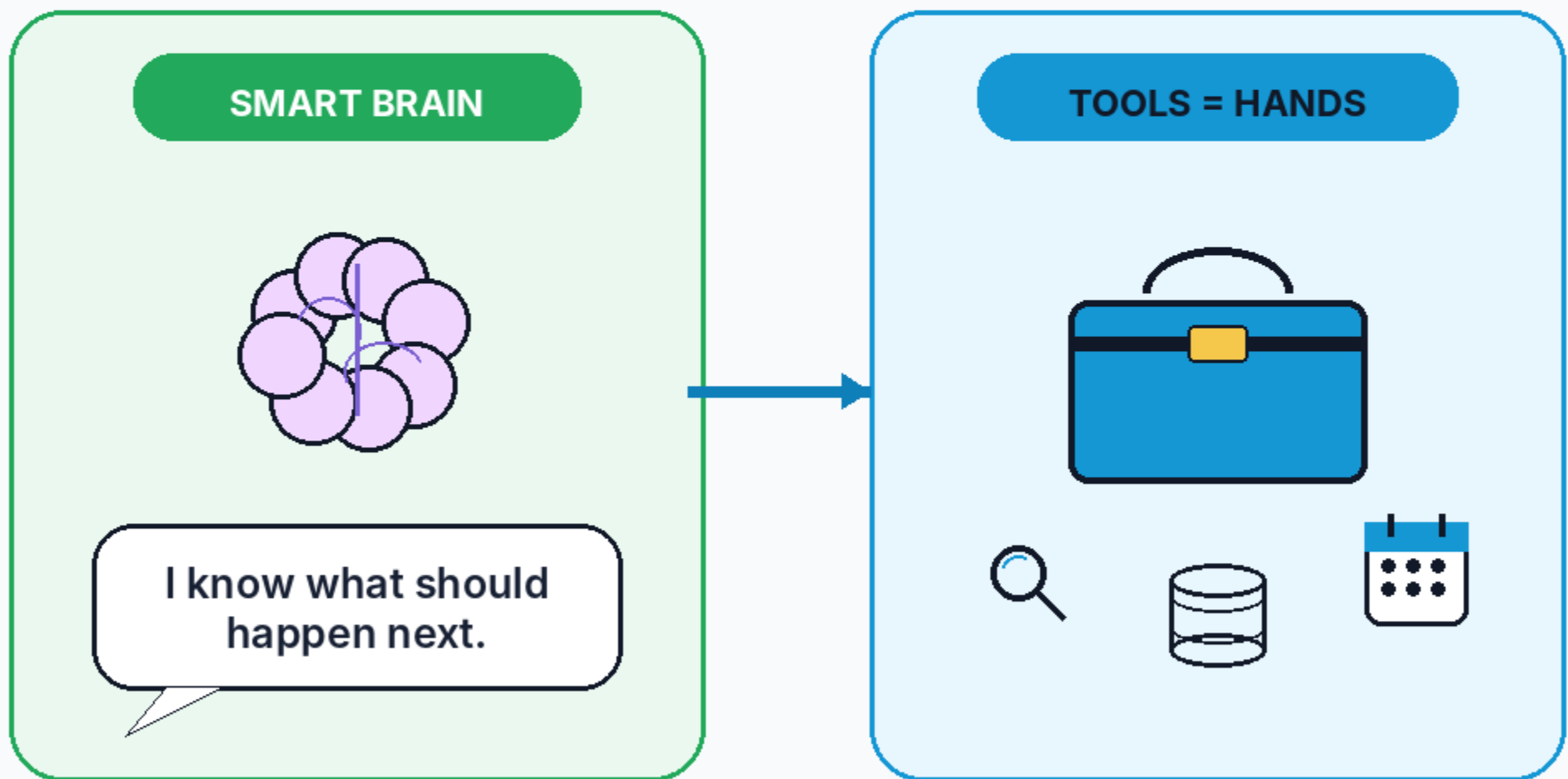


HOW AI AGENTS USE TOOLS

Function calling—explained without jargon.

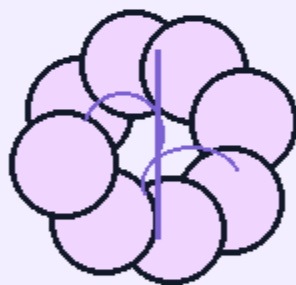


The model chooses the tool. Your application performs the action.

A SMART BRAIN STILL NEEDS HANDS

An LLM can understand a request and decide what should happen. But understanding alone cannot touch the outside world.

THE MODEL



"The next step is to check the live order status."

THE TOOLS



LOOK UP • SEND • UPDATE

Tools turn useful words into useful work.

KNOWING IS DIFFERENT FROM DOING

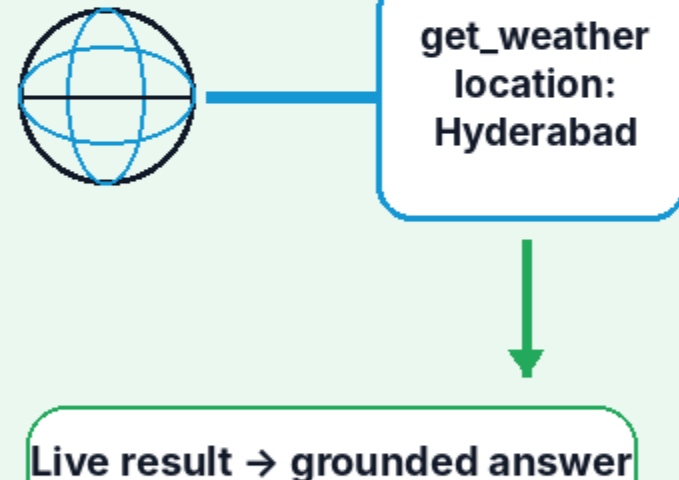
The model can answer from its context. A tool supplies live data or performs a real action.

WITHOUT A TOOL



The model may explain weather, but it cannot confirm live conditions from memory alone.

WITH A WEATHER TOOL



The agent uses the returned conditions to answer—or take the next step.

Live information and real actions come from tools—not model memory.

WHAT COUNTS AS A TOOL?

A tool is one specific capability that your system makes available to the model.



WEB SEARCH

Live information



DATABASE

Orders, users, inventory



**CALCULATOR /
CODE**

Math and analysis



**CALENDAR /
EMAIL**

Schedule and
communicate



FILES / RAG

Read trusted documents



BUSINESS API

Create or update records

Best practice: give every tool one clear job.

FUNCTION CALLING, IN PLAIN ENGLISH

Also called tool calling: the model converts a normal request into a structured request for a named tool.

1

USER SPEAKS NORMALLY

"Find my order and start a return."



2

MODEL CREATES A TOOL REQUEST

`lookup_order(order_id="A102")`



3

APPLICATION RUNS THE FUNCTION

Result: delivered yesterday

Natural language → structured tool request → executed result

EVERY TOOL NEEDS A CLEAR MENU CARD

The model chooses tools using their name, description, and expected inputs.

TOOL DEFINITION

Name

find_restaurants

Description

Find restaurants that match the user's location and preferences.

Inputs

city

text

minimum_rating

number

outdoor_seating

yes / no

UNCLEAR

do_stuff(data)



CLEAR

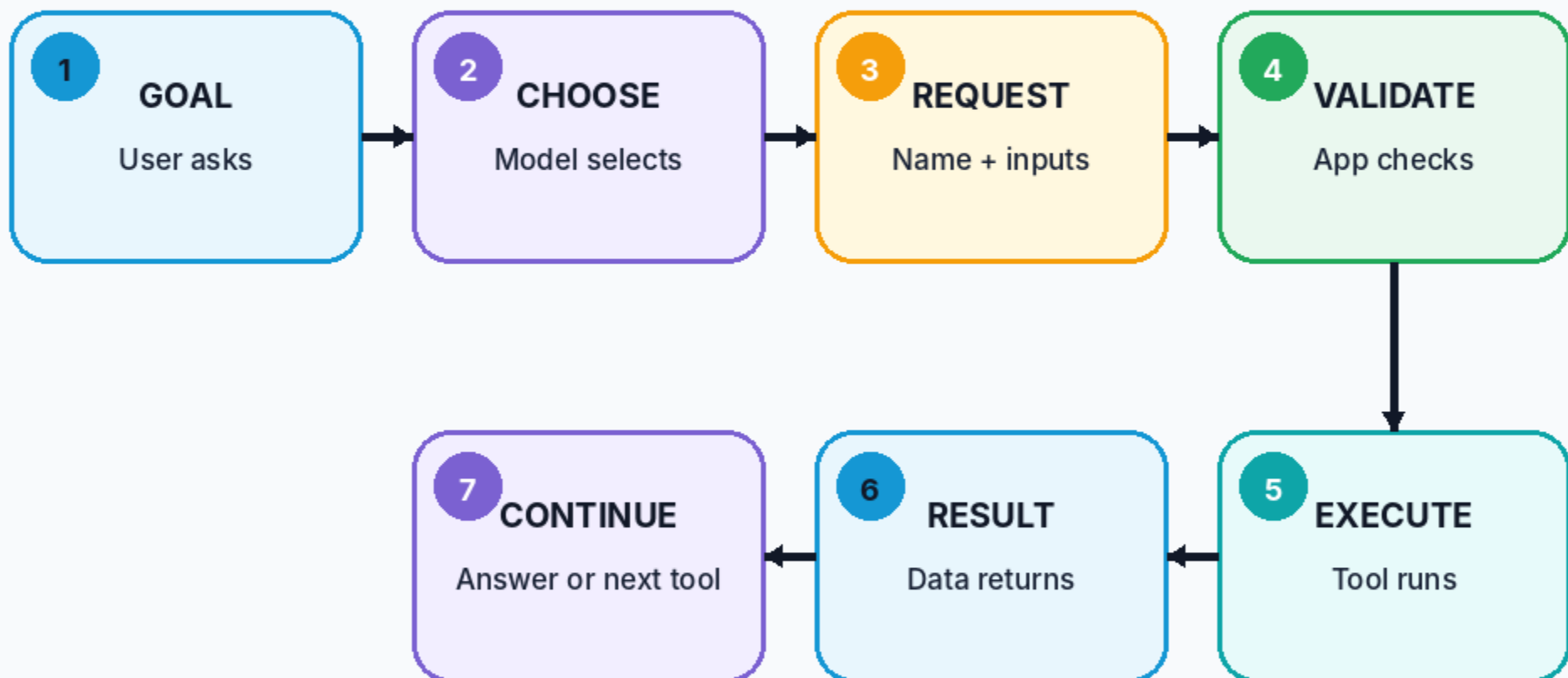
find_restaurants(city,
rating, seating)



Clear tool definitions lead to better tool selection.

THE TOOL-USE HANDSHAKE

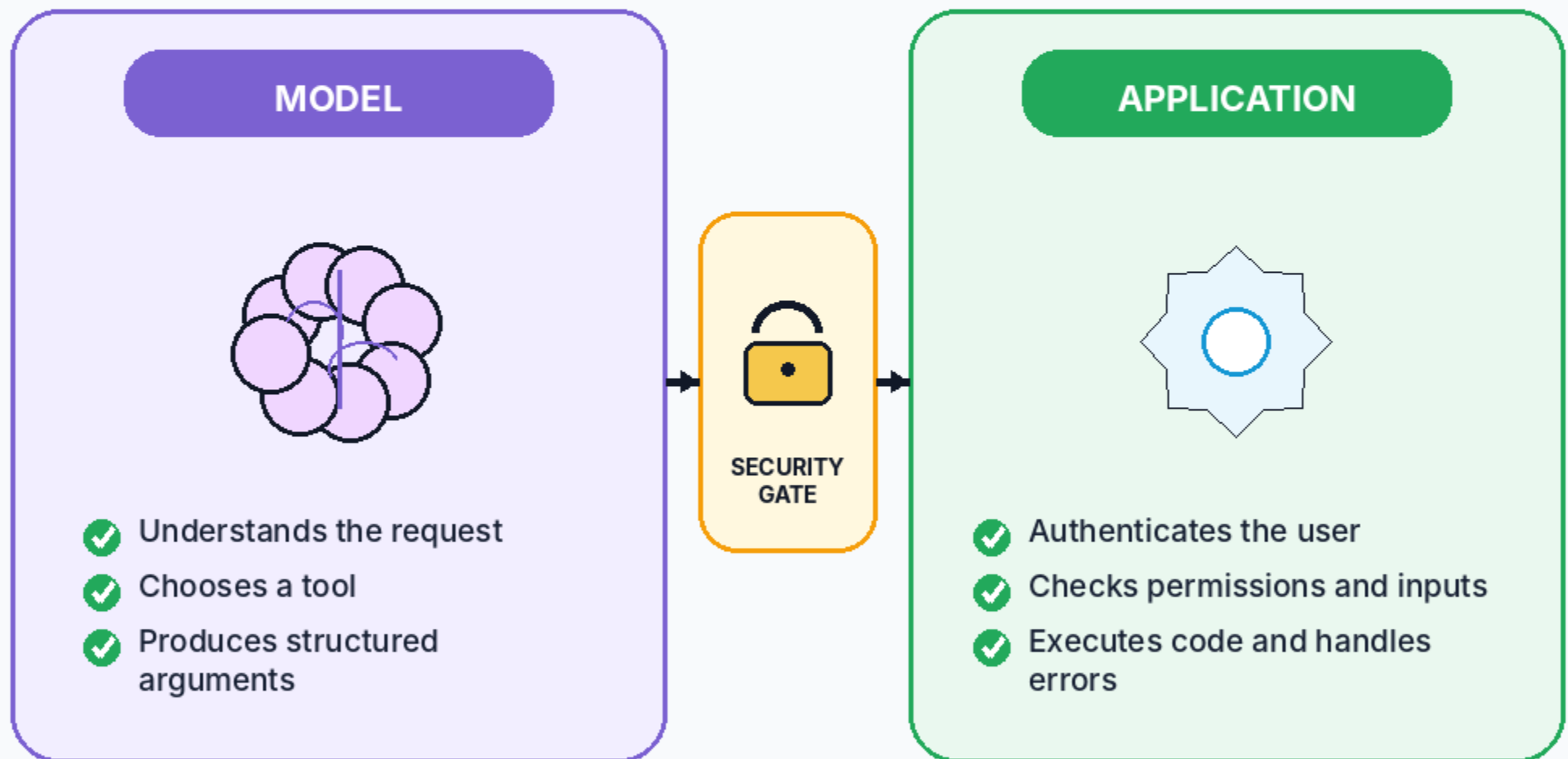
A safe tool call is a conversation between the user, model, application, and external system.



The loop may happen once—or repeat until the goal is complete.

THE MODEL IS NOT YOUR BACKEND

The model proposes an action. Trusted application code decides whether and how it runs.



Never let generated arguments bypass your normal security rules.

EXAMPLE 1:

FIND A RESTAURANT

One user goal becomes several structured tool calls.



"Find a biryani place near Hitech City, rated 4.3+, with outdoor seating."

1

search_places



query + location

2

get_reviews



minimum rating

3

get_distance



current location

Result: a shortlist that satisfies the filters—without the user specifying every step.

EXAMPLE 2:

SCHEDULE A MEETING

Read first. Propose a choice. Write only after approval.

"Book 30 minutes with Priya next week."



1 **find_contact**
Resolve which Priya

2 **check_calendars**
Find mutual free time

3 **propose_slot**
"Tuesday at 3 PM?"

4 **USER APPROVES**
Explicit confirmation

5 **create_event**
Now write to calendar

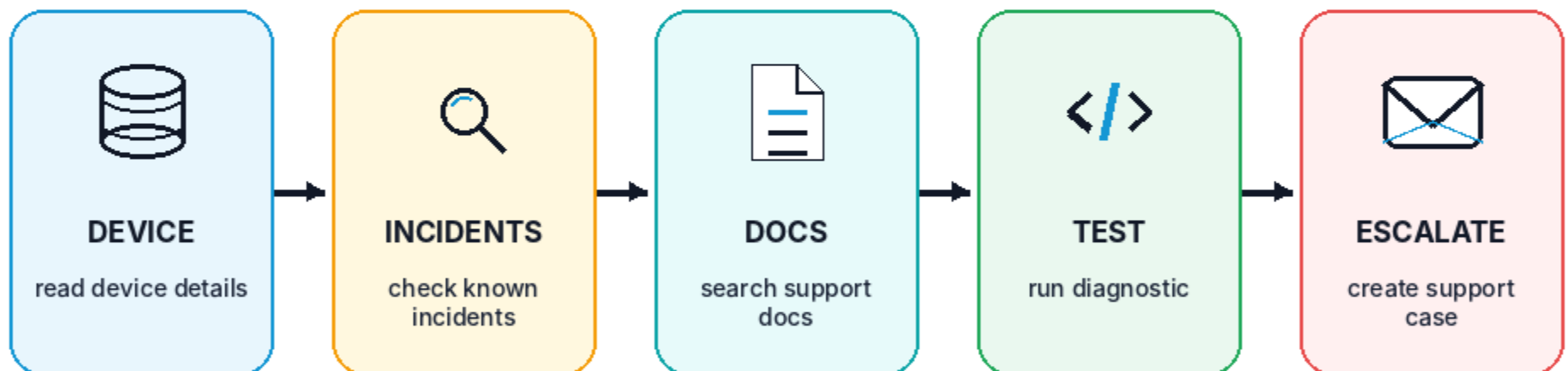
Actions with side effects—sending, booking, paying, deleting—often need confirmation.

EXAMPLE 3:

CUSTOMER SUPPORT

A good support agent investigates with tools instead of confidently guessing.

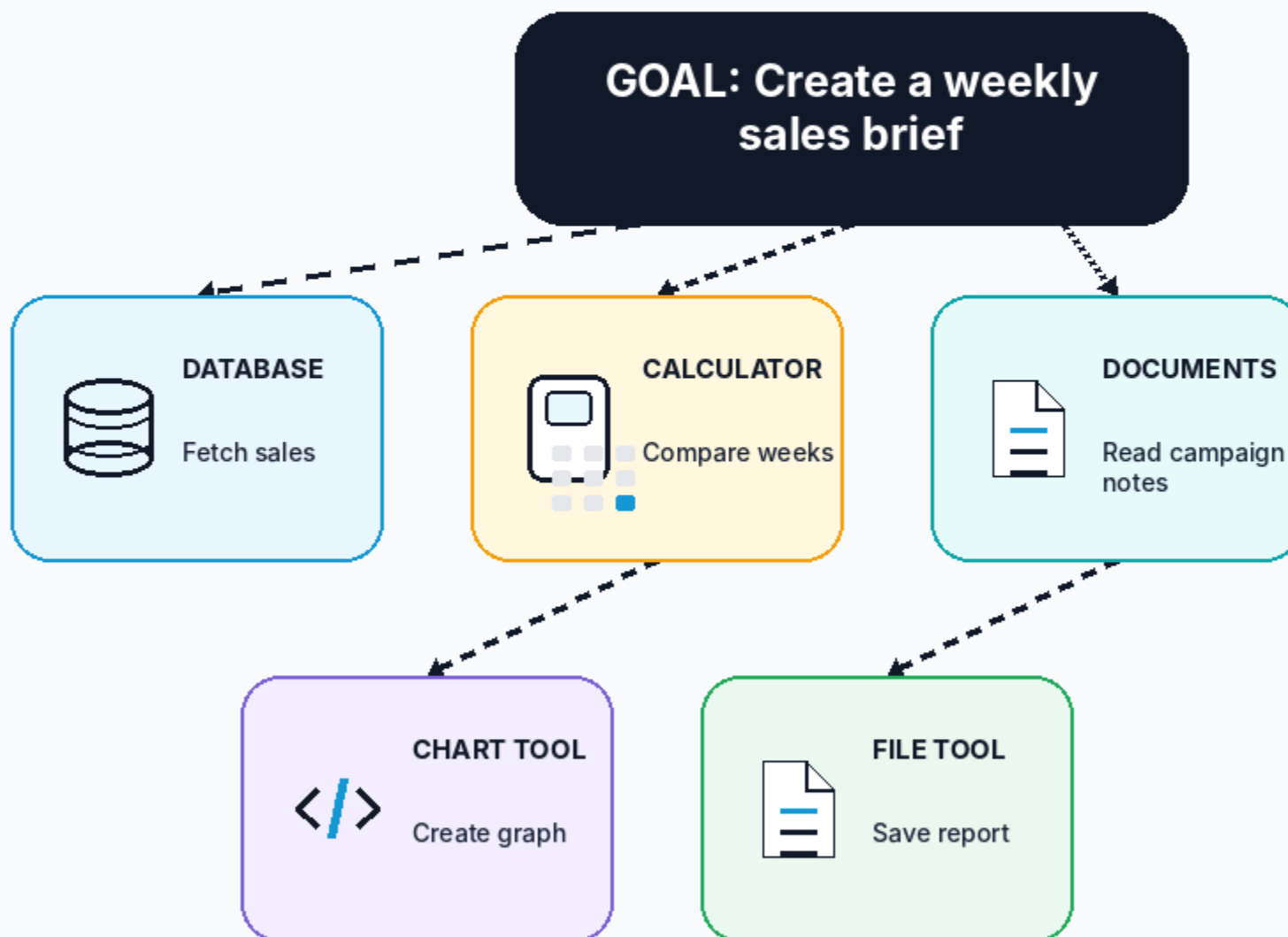
"My headphones stopped working after yesterday's update."



Each tool adds evidence. If evidence is insufficient, the agent escalates.

ONE GOAL CAN NEED MANY TOOLS

The agent coordinates the work. Each tool remains small and specialized.



Some tools can run in parallel; others must wait for earlier results.

HOW DOES THE AGENT CHOOSE A TOOL?

Tool selection is guided by the goal, context, descriptions, inputs, results, and permissions.

WHAT IT CONSIDERS

- ✓ The user's goal
- ✓ Tool names and descriptions
- ✓ Required inputs
- ✓ Current conversation context
- ✓ Previous tool results
- ✓ Rules and permissions

EXAMPLE: LEAVE BALANCE

"How many leave days do I have?"



search_policy

Explains rules



get_leave_balance

Reads this user's balance



request_leave

Creates a leave request

Good descriptions make the correct tool easier to choose.

WHEN TOOL CALLS GO WRONG

Failure is normal. A reliable agent must recognize it and respond safely.



WRONG TOOL

Searches the web when the answer is in the customer database.



BAD ARGUMENTS

Missing city, invalid order ID, or unsupported value.



TOOL FAILURE

Timeout, rate limit, unavailable API, or malformed result.



UNSAFE ACTION

Sends, pays, changes, or deletes without confirmation.

ASK → VALIDATE → RETRY → FALL BACK → ESCALATE

A SAFE TOOL CHECKLIST

Give the agent useful power—but keep that power bounded.



BOUNDED POWER

- ✓ Authenticate the user
- ✓ Grant least-privilege access
- ✓ Validate types and allowed values
- ✓ Ask before money, messages, changes, or deletion
- ✓ Set timeouts, retry limits, and budgets
- ✓ Log tool calls and results
- ✓ Provide a clear human fallback

A tool is trustworthy when its permissions, inputs, and failure behavior are controlled.

FUNCTION CALLING CONNECTS LANGUAGE TO ACTION

The model chooses a named tool and creates structured inputs. Your application validates and executes the action, returns the result, and the agent decides what happens next.

[NEXT](#)

DAY 7: HOW AGENTS PLAN

ReAct vs. Plan-and-Execute

What is the first tool you would give your AI agent?

