上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

# Lecture 37: Autonomous Systems

## Multi-Agent Systems & Human-AI Collaboration

Tao Huang

John Hopcroft Center, School of Computer Science, Shanghai Jiao Tong University

https://taohuang.info/cs3317

https://oc.sjtu.edu.cn/courses/89538

AI tools assisted in generating some figures in these slides. All such content has been reviewed, and the instructor is responsible for its accuracy.

# Where We Are

*L36 made one agent that can sustain a long task.*
*L37 asks: what if one agent isn't enough?*

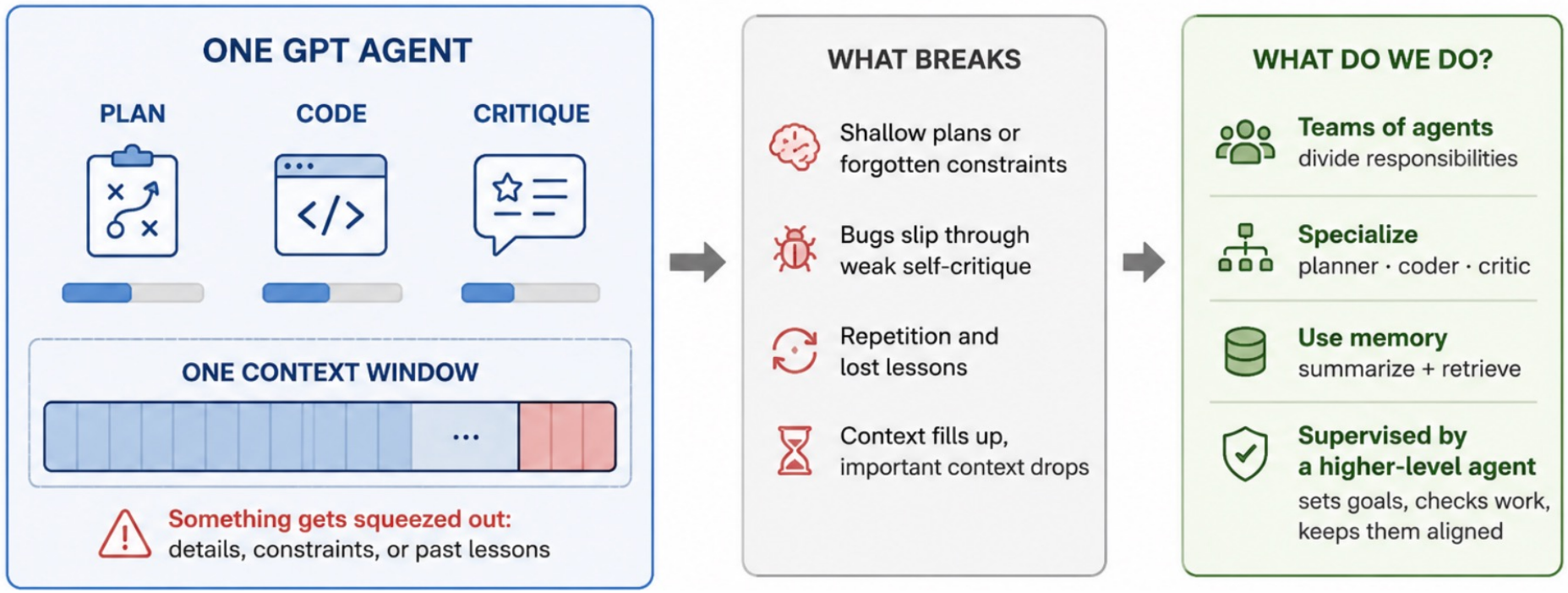
**Two threads we left open last time:**

- Can several **specialized agents** outperform one generalist?
- How do **humans stay in the loop** when agents run for hours?

*The arc: L35 = the loop · L36 = the long horizon · L37 = many agents + the human*

# The Question

*A single GPT can plan, code, and critique — but never all three well, at once, in one context window. What breaks, and what do we do about it?*



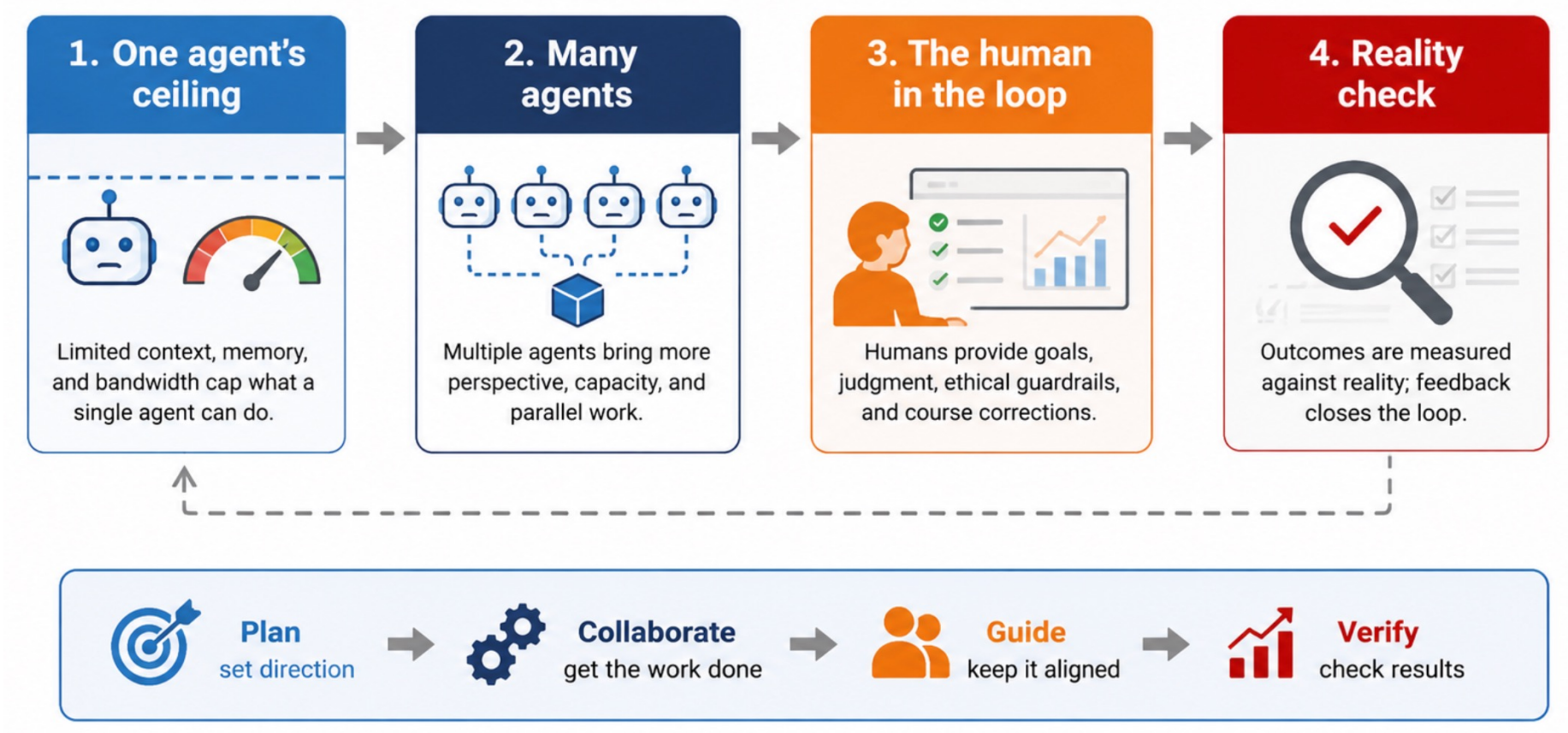

**Today: the limits of one agent → teams of agents → who supervises them.**

# Objectives

*By the end of this lecture, you should be able to:*

- **Explain** why a single agent saturates as tasks scale (role, context, parallelism).
- **Distinguish** the main multi-agent patterns (orchestrator-worker, pipeline, debate, peer).
- **Evaluate** whether a multi-agent gain comes from the design or just from more compute.
- **Place** a deployed system on the autonomy spectrum and locate where a human must stay.
- **Critique** the cost, trust, and accountability tradeoffs of autonomous systems.

# Roadmap

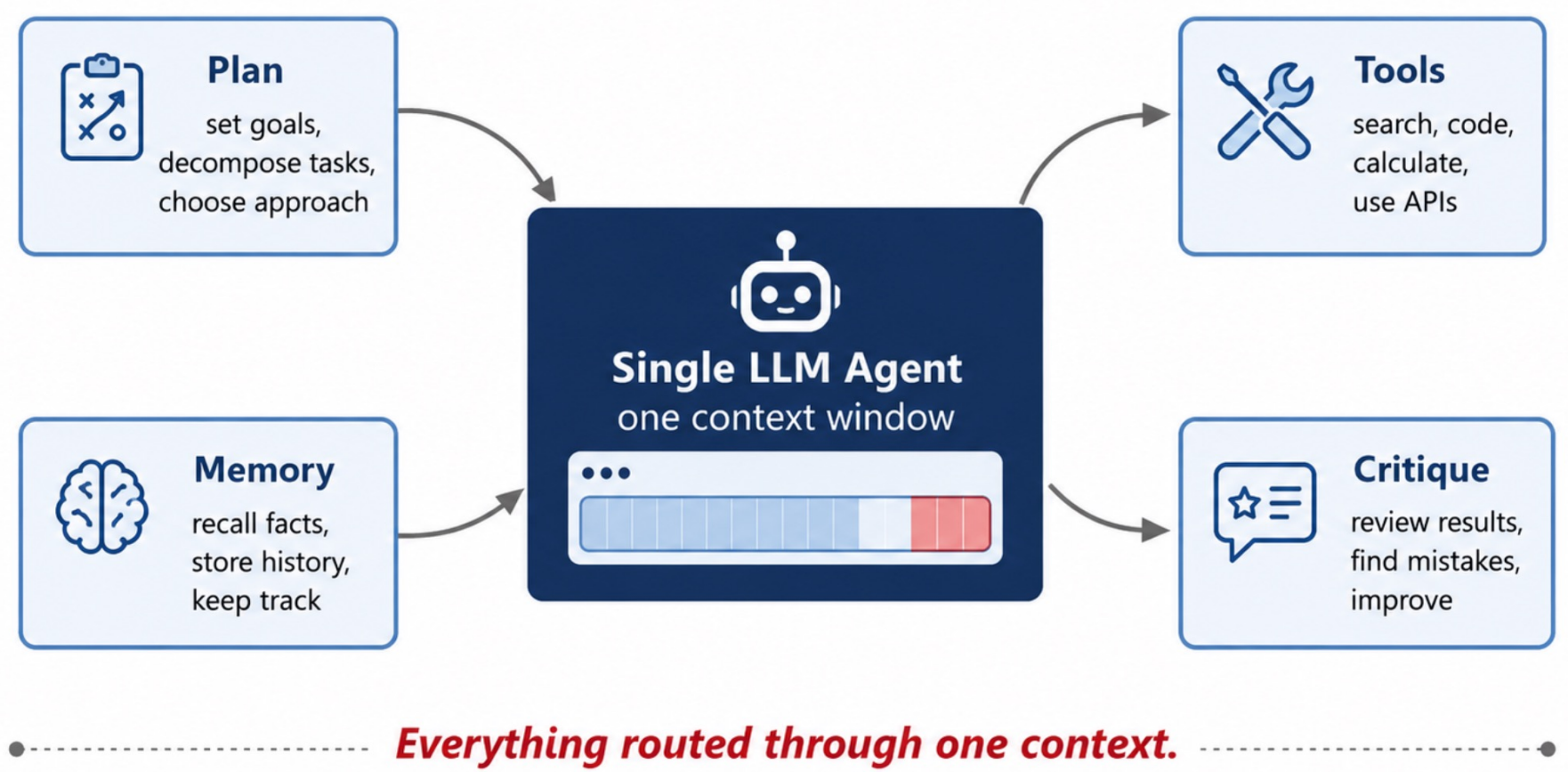



*Four beats — each one forced by the limit of the one before it.*

# 1. Why One Agent Isn't Enough

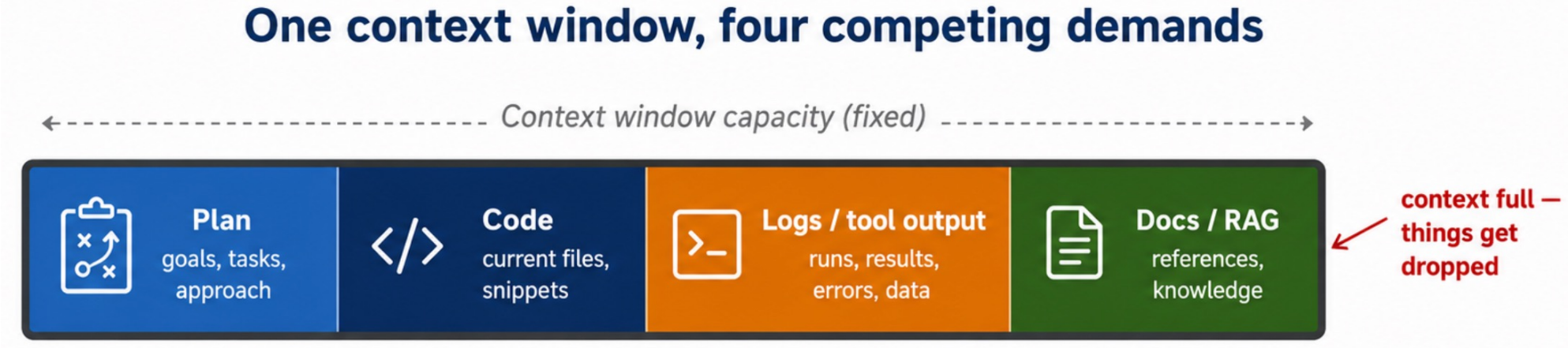
# The Single-Agent Ceiling

- **Situation:** our L35–L36 agent — one model, one context window, a ReAct loop, memory, and tools. It works. Until the task gets big.



# Three Things That Saturate

- **Complication:** one agent hits three walls at once.




- **Context overload** — plan, code, logs, docs all fight for one window (recall L36's auto-compact).
- **Role conflict** — the same prompt can't be a rigorous planner and a skeptical critic.
- **No parallelism** — one agent does one thing at a time; a 10-file refactor is serial.

# Think: Where Does This Break First?

*You ask ONE agent to: research a topic, write a 20-page report, then fact-check its own claims. Which step fails first — and why?*

(a) the research

(b) the writing

(c) fact-checking its own work

(d) it runs out of context first

*Discuss with your neighbor before we reveal it.*

# The Move

**Question:** *if one context, one role, one thread is the bottleneck — what's the obvious move?*

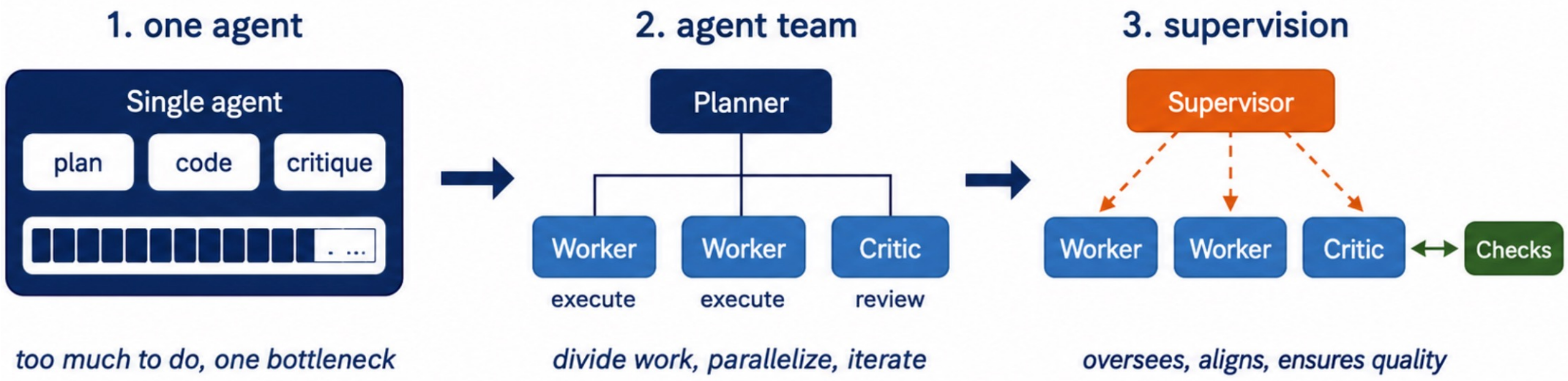
*Answer: split the work across several agents*
*— each with its own role, context, and tools — and let them talk.*

**We don't make the agent smarter. We make a team.**

# 2. Multi-Agent Systems

# What Is a Multi-Agent System?

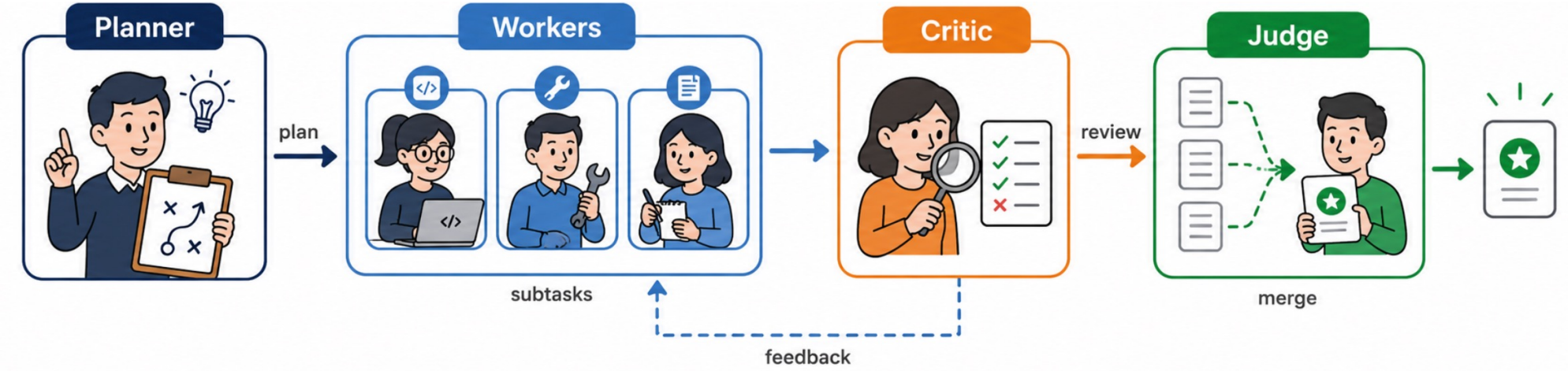
**Definition:** several LLM agents — each with a role, its own context/memory, and possibly its own tools — that communicate via messages to solve a shared task.



# Roles & Specialization

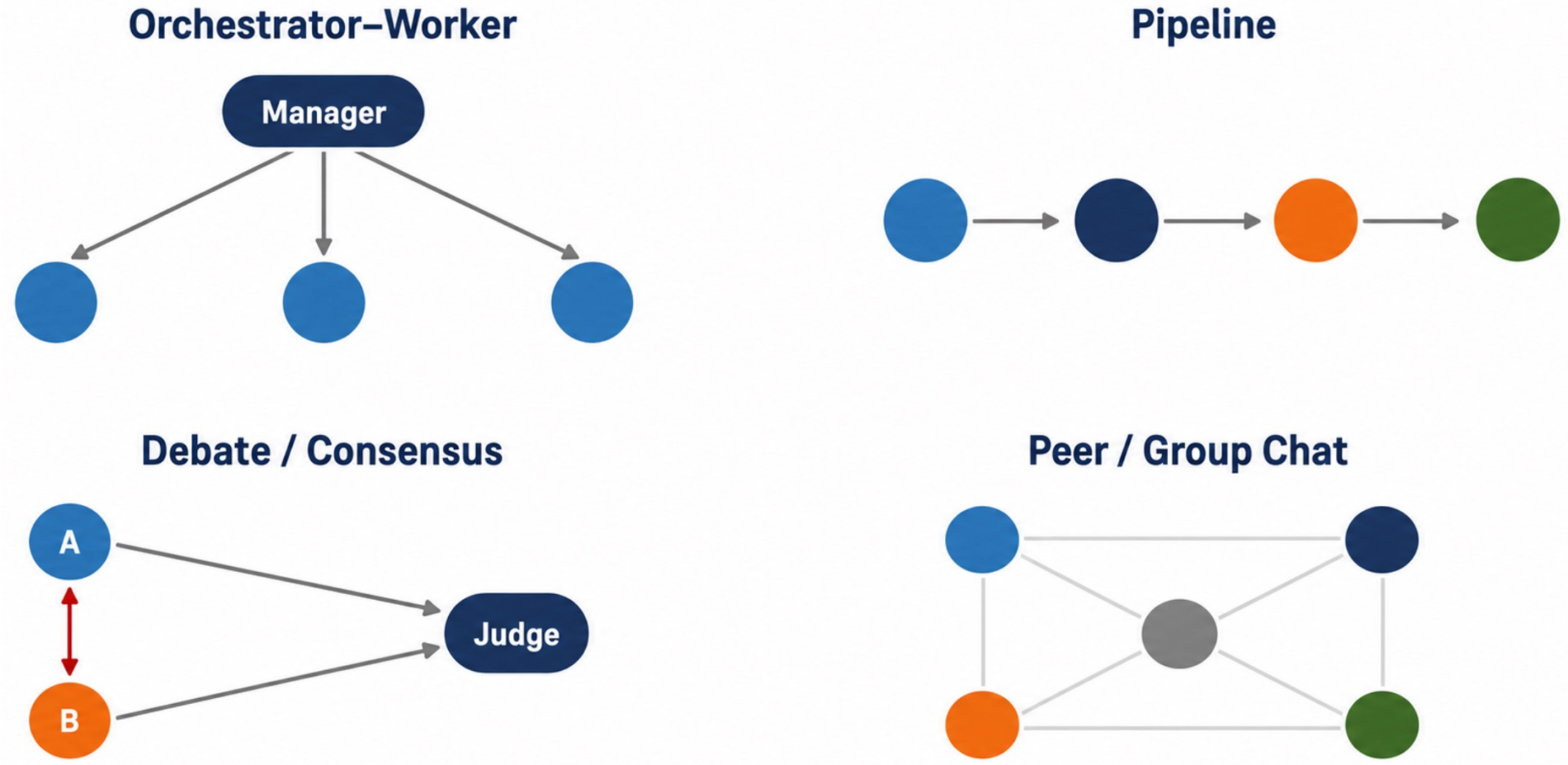
**Common roles:**

- Planner / Orchestrator — decomposes and delegates.
- Workers / Specialists — each handles one subtask.
- Critic / Reviewer — a separate reader that checks the work.
- Judge / Aggregator — picks or merges the final answer.




*A critic with its own context is a genuinely different reader — it catches what the author missed. This is 'separation of concerns', applied to prompts.*

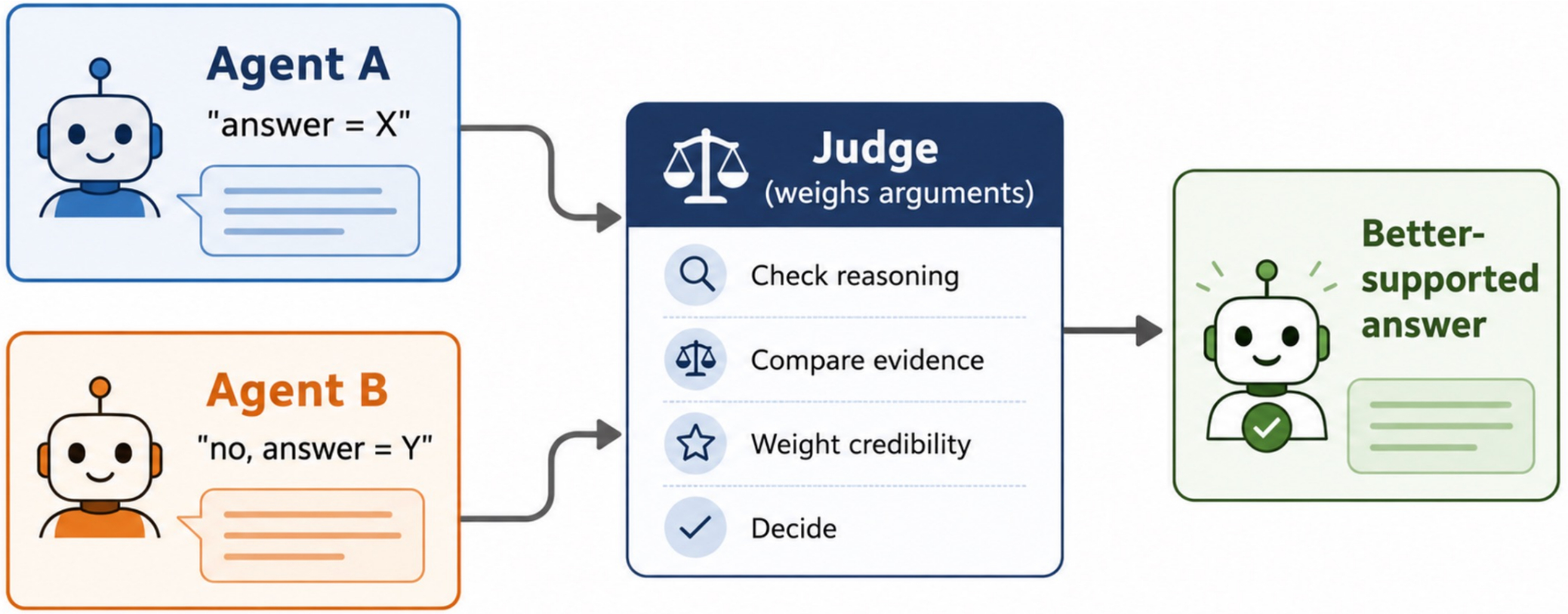
# Coordination Topologies




*Manager-led · assembly line · argue-then-judge · free-form channel.*

# Why Debate Works (a Little)

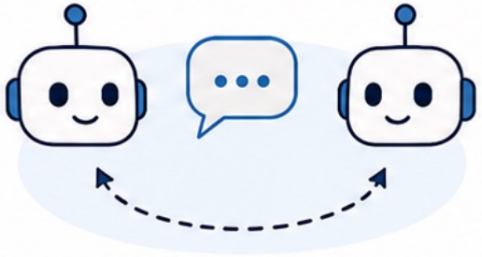
Multiple agents arguing surfaces errors a single chain-of-thought hides; a judge picks the better-supported answer.



# The Systems Zoo

## AutoGen

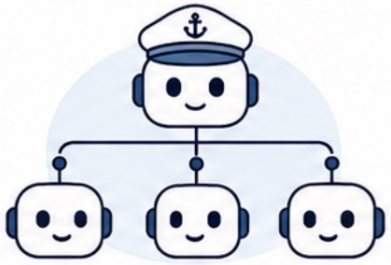
*(Microsoft)*



Conversational multi-agent framework

## CrewAI

*(open source)*



Role-based agent 'crews'

## MetaGPT / ChatDev

*(research)*

A software company of agents

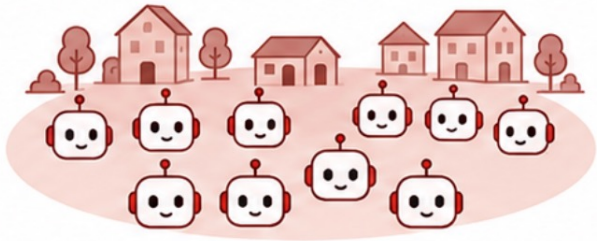
## CAMEL

*(research)*

Two agents role-play a task

## Generative Agents

*(Stanford)*



25 agents living in a sandbox town

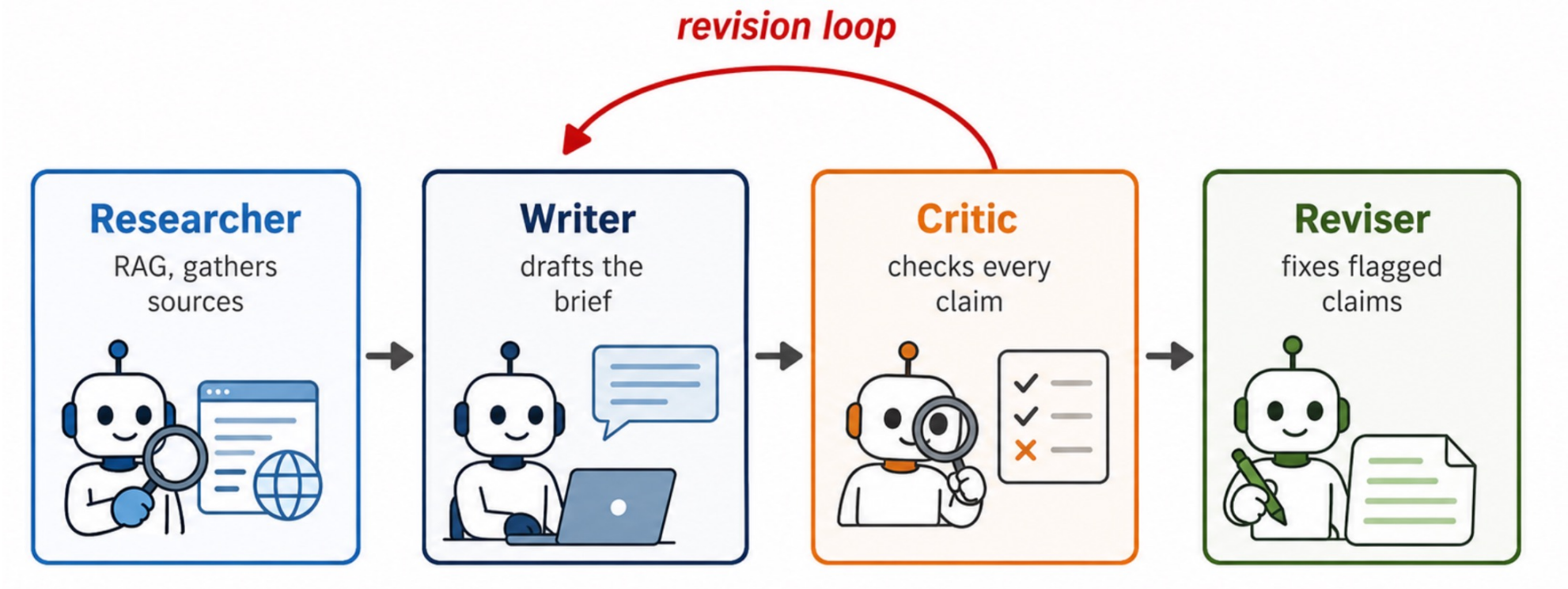
# Emergent Social Behavior: “Smallville”


Park et al., Generative Agents: Interactive Simulacra of Human Behavior, 2023

*An agent plans a party; invitations propagate through the town.*
*Believable — but be careful reading too much into it.*

# Worked Example:  the Research-Report Crew

- **Task: produce a vetted 2-page brief on solid-state batteries.**




*Walk it live: the Critic flags an unsupported claim → the Reviser fixes it → the Judge approves. One loop is usually enough.*

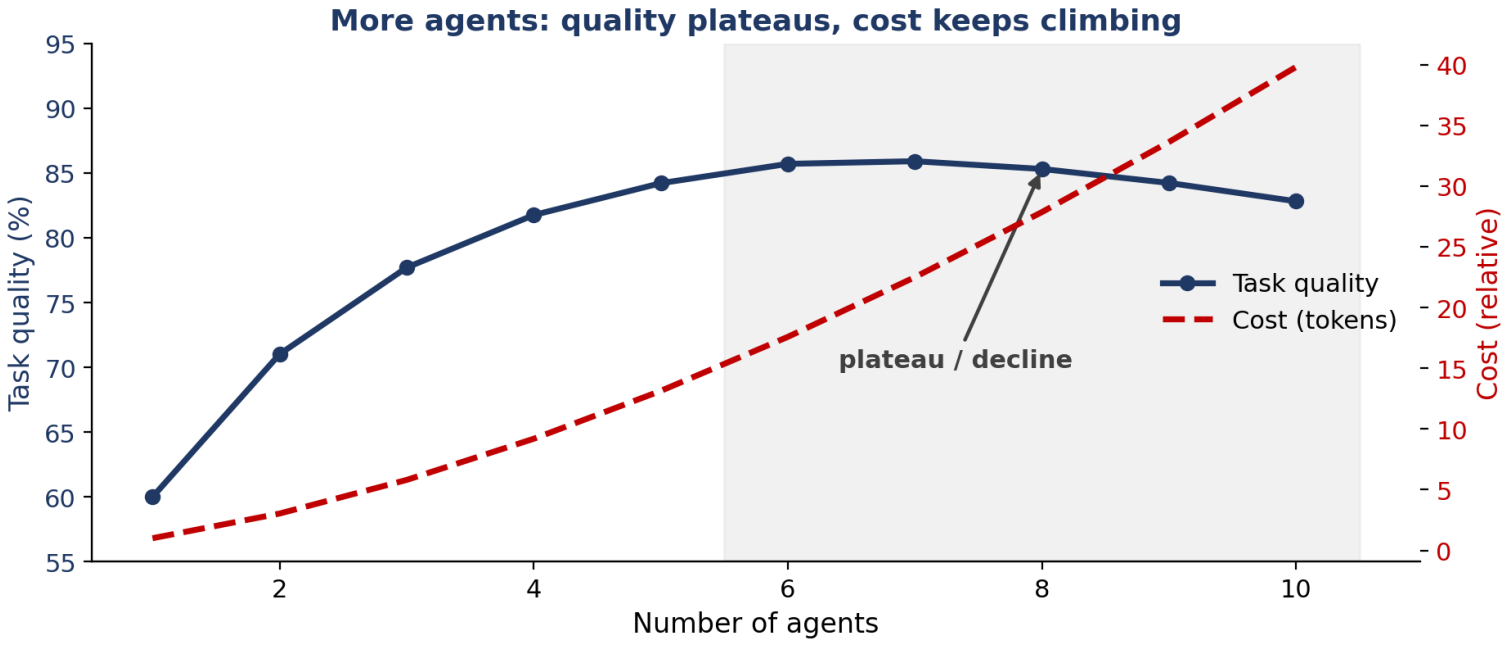
# Does “More Agents” Actually Help?

**The seductive story:**

- more agents = more brains = better.

**The reality:**

- Cost explodes (N × M × tokens)
- Errors propagate between agents
- Echo chambers — same model agrees
- Coordination overhead, deadlocks



# 3. Human-AI Collaboration

# The New Situation

- **Situation:** whether one agent or a crew, they now run **autonomously, for a long time, taking real actions**
  — writing files, sending emails, spending money.

**L36's long horizons + L35's tools = real-world consequences.**

# The Control Problem

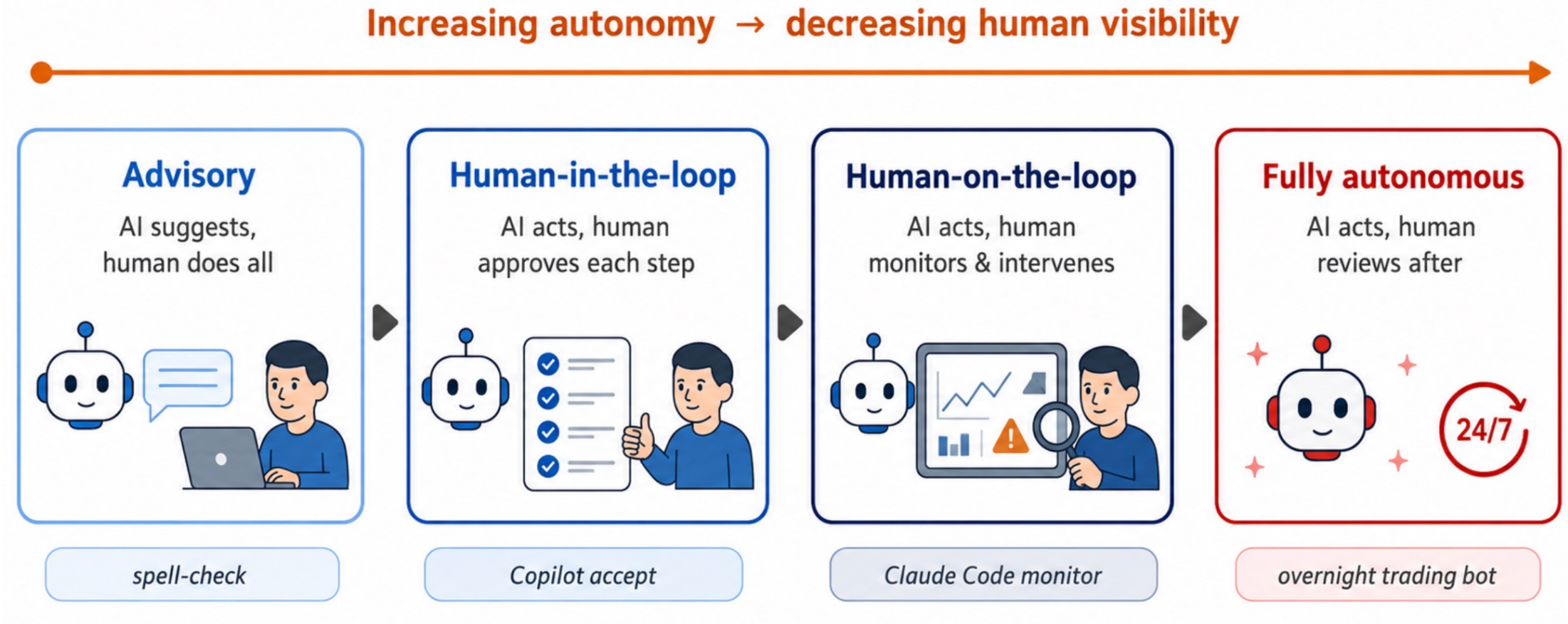
**Complication:** as autonomy rises, human visibility drops and errors compound silently.

- High-stakes, irreversible actions (delete, deploy, pay) cannot be fully delegated.
- Yet babysitting every token defeats the point of an agent.

*Question: how do humans stay meaningfully in control — without approving every single step?*

# The Autonomy Spectrum

- Answer (part 1): autonomy is a ladder, not a switch.



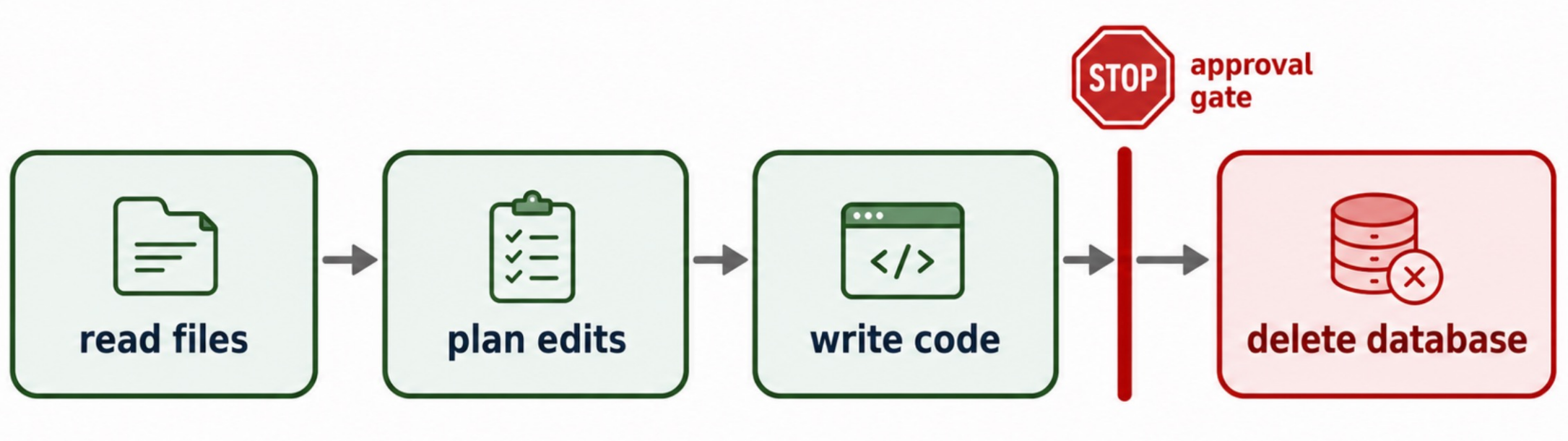
# Collaboration Modes

Answer (part 2): the same agent can sit in different modes depending on the task.

- Tool — autocomplete; you drive entirely.
- Assistant — Copilot suggests, you accept or reject.
- Collaborator — pair-programming; mixed initiative.
- Supervised operator — Claude Code runs, asks before destructive ops.

*Mixed-initiative: either party can take the next step;*
*a good interface makes the handoff cheap.*

# Oversight Mechanisms

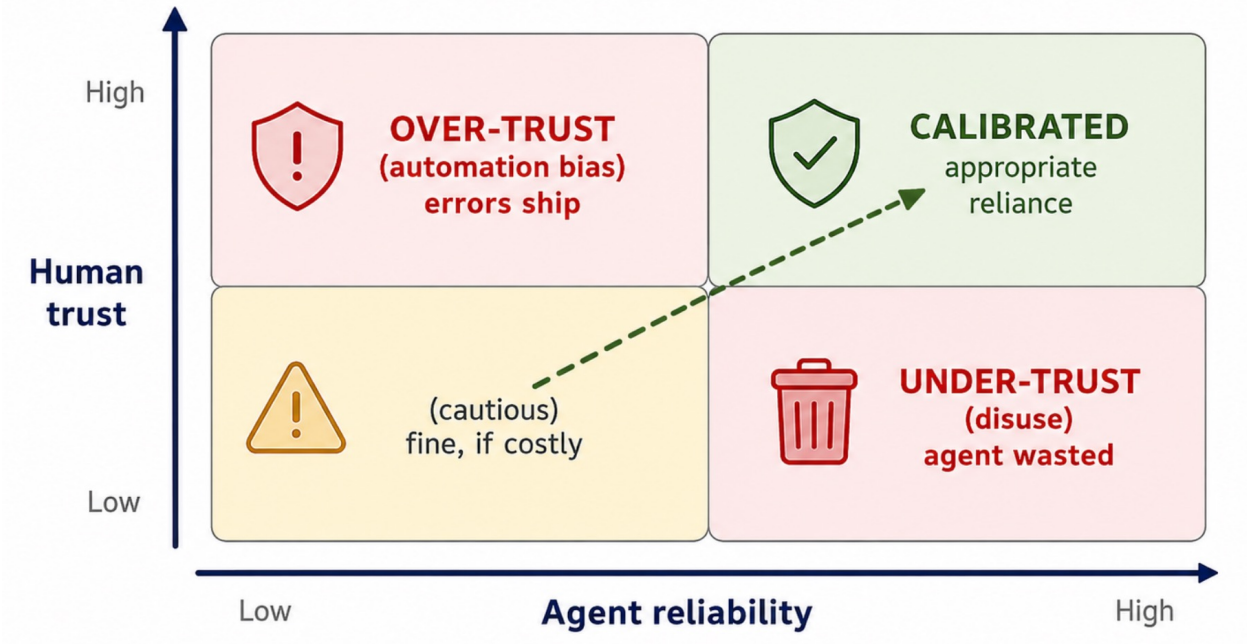



- **Approval gates** for irreversible actions.
- **Sandboxing** — let it fail safely.
- **Audit logs / traces** — see why it did X.
- **Interpretable plans** — show before executing.

# A Real Approval Gate


*The approval gate is the cheapest, highest-leverage safety mechanism in deployed agents today.*

# Trust Calibration




**Two failure modes:**

- **Over-trust** → automation bias: you stop checking; the error ships.
- **Under-trust** → disuse: you redo everything; agent wasted.

**Goal:** trust scaled to demonstrated reliability.

# Collaborations in the Wild

- Coding — Copilot / Cursor / Claude Code: suggest, human accepts.
- Radiology — AI flags suspicious regions, the doctor decides.
- Content moderation — AI triages volume, humans handle edge cases.
- Customer support — AI drafts the reply, a human sends it.

*The pattern across all of them:*
*AI handles volume, humans keep judgment and accountability.*

# Where Autonomous Systems Deliver

- **Works now:**
  bounded, reversible, high-volume, human-reviewable tasks — drafting, triage, scoped coding.

- **Still shaky:**
  open-ended autonomy, high-stakes irreversible actions, and 'multi-agent wins' that disappear once you compute-match the baseline.

*The bottleneck isn't capability demos — it's trust, cost, and accountability.*

# Open Problems / Frontier

- Accountability gap — when an agent crew makes a bad call, who is responsible?
- Evaluation — how do we honestly measure a multi-agent system at all?
- Cost & latency — autonomy is expensive; many agents, many turns.
- Safety as autonomy grows — the more it acts, the more an oversight failure costs.

# Summary

- One agent saturates on role, context, and parallelism — so we build teams.
- Coordination helps (orchestrator, pipeline, debate) — but often less than the token bill suggests. Always compute-match before believing a win.
- Autonomy is a ladder — the human's job is judgment, accountability, and the approval gate.
- Trust must be calibrated — over-trust ships errors, under-trust wastes the agent.

# Next Week

*Every agent so far — one or many — acts through text and software.*
*L38 asks: what happens when the agent has a body?*

**Two questions for next lecture:**

- How does an agent turn **pixels and language into motor actions**?
- Why is the **physical world** so much harder than the digital one?

**L38: Perception-to-Action Models — the Vision-Language-Action paradigm.**
**(Week 14, Embodied Intelligence)**