

↳ Interview Prep points → #

↳ What is Multi Agent? $\xrightarrow{\text{MAS}}$

↳ Why Multi Agent?

↳ Static vs dynamic Role?

↳ Patterns in Multi Agent

↳ Lerngraph vs Crew AL

↳ Components of Crew AL
 $L, +$

↳ Stock analyzer (new AL)

he ~ AZ

AZ

Data Science
AI

{year}

ML was new

+

in India Job market

he ~ AZ Agents

{fundamental
core topic
why doing
=}

core fundamental

ML =

regression"

using

configur

Agen \propto^2

\equiv

$\uparrow 5\%$

~~-T~~

Lm $\left\{ \begin{array}{l} \text{... Educated poor} \\ \text{less} \end{array} \right.$

↳ genAI authors / Agents

how to improve LLM?

↳ Components to RAH? ↴

↳ Case Studies on RAH?

↳ how to improve RAH → ?

|| ↳ Agents vs genAI ?.

↳ how to create chunks ?

↳ how all this is working ?

| change →

| ↓

enk →

out =

↓

vector

= ↓

DB ↪

enk

↓

vector

↓

similarity

, kNN

↓

LLM

↓

gen

↳ Prompt finetuning
= $\text{D}_{\text{fin}} + \text{fin} + \text{User}$
↳ REACT

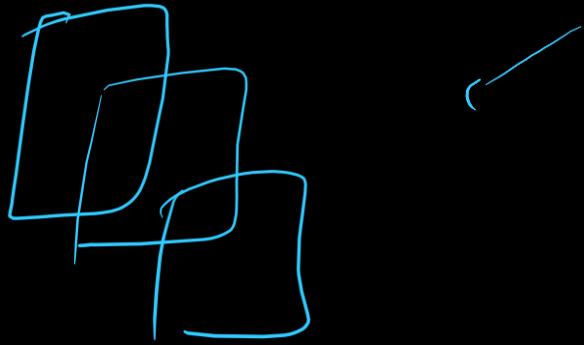
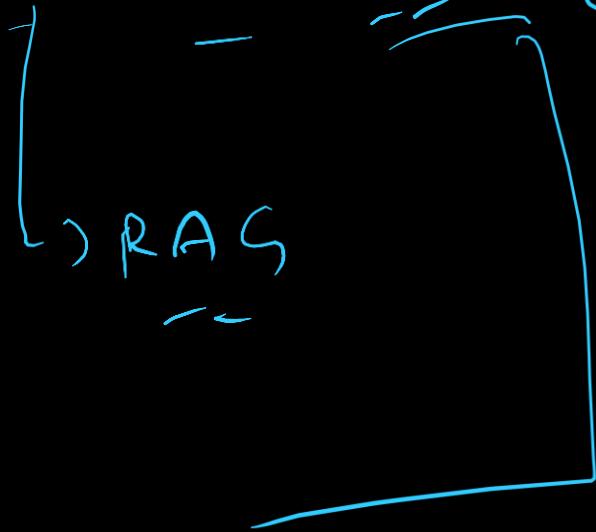
↳ fine tune
↳ LLM \rightarrow 100Billion + $\sqrt{100k}$ param = 8LORA

↳ RAG
↳ [] \rightarrow LLM + 1
↳ knowledge

↳ tool call / find

By which you can improve our LLM

Recommender systems

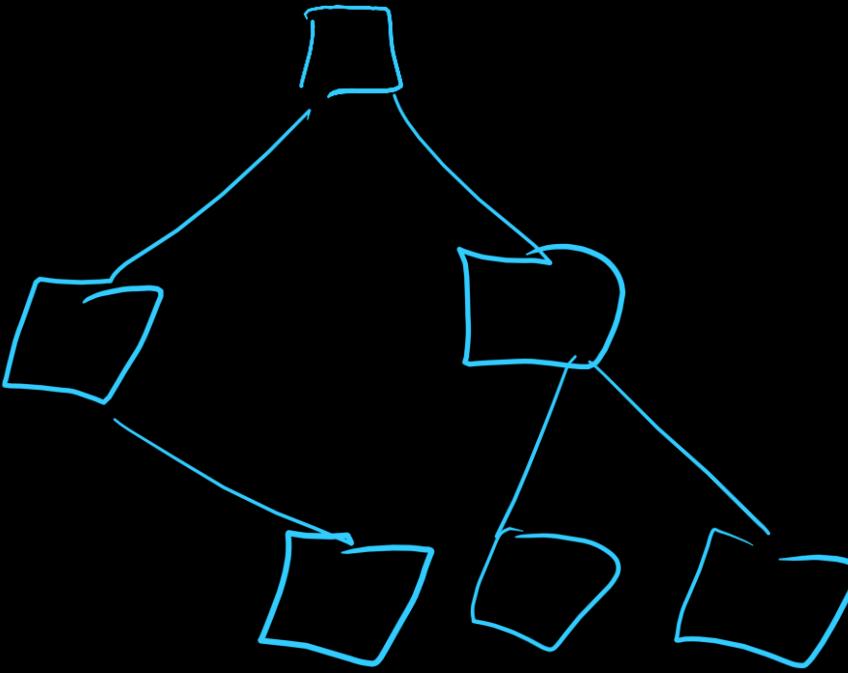


RAS

Now, every knows almost the
same amount about

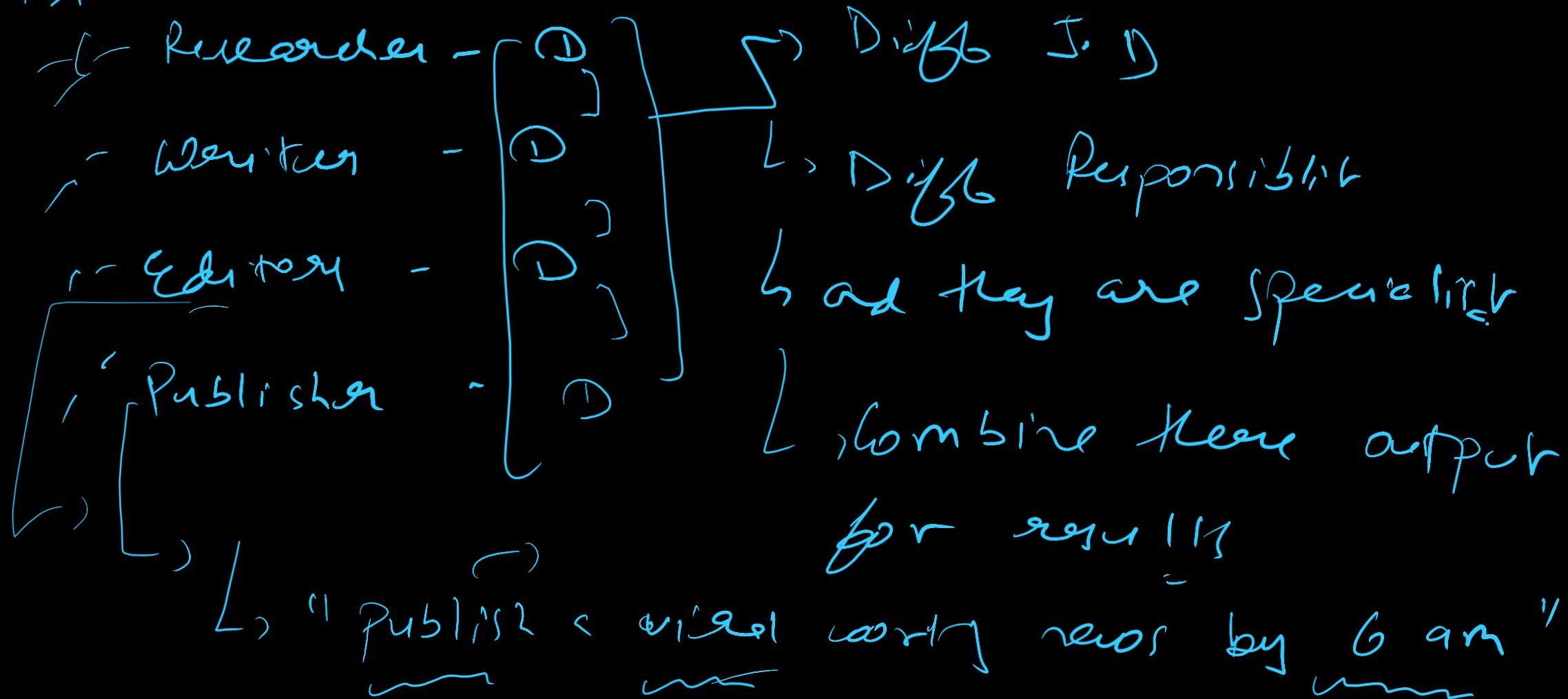
$$\boxed{98\%}$$
 $\hookrightarrow 2\%$







Multi Agent System



What is a Multi-Agent System?

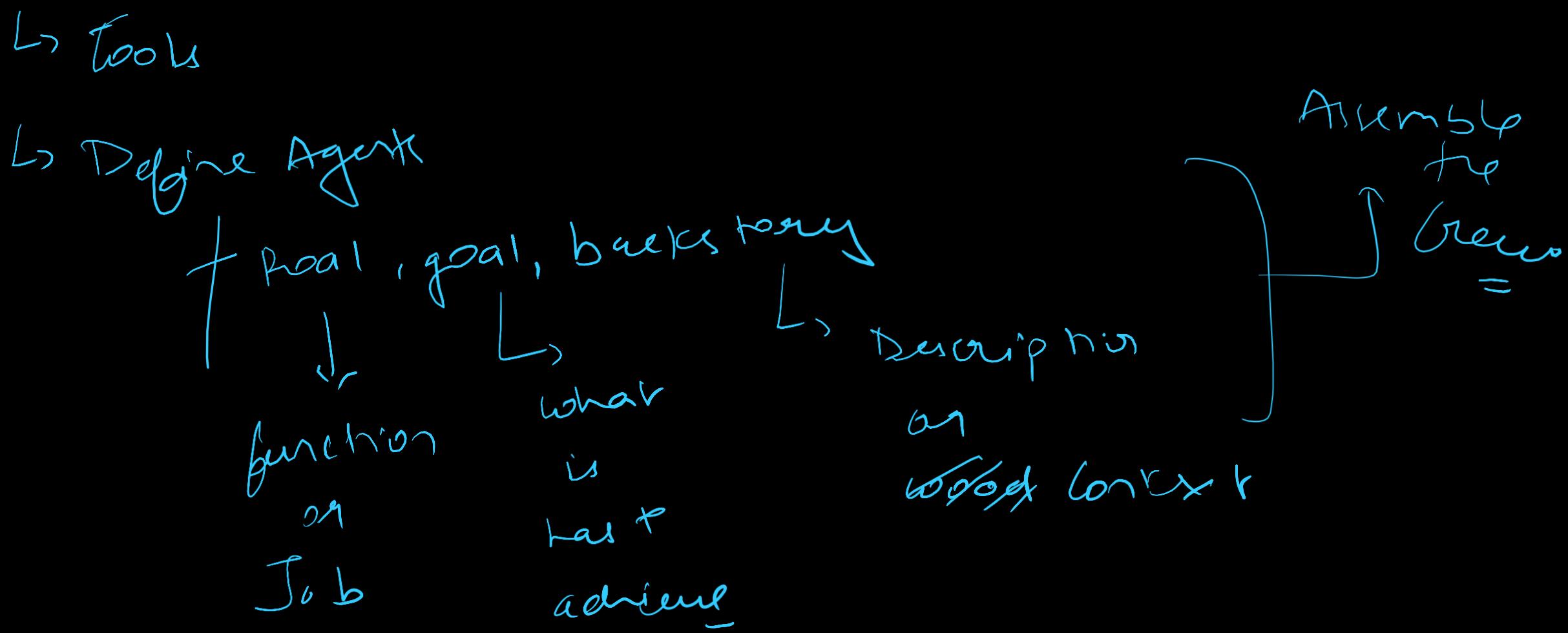
• **Definition:** A multi-agent system (MAS) is a collection of autonomous agents that interact within an environment to achieve goals.

• Core characteristics:

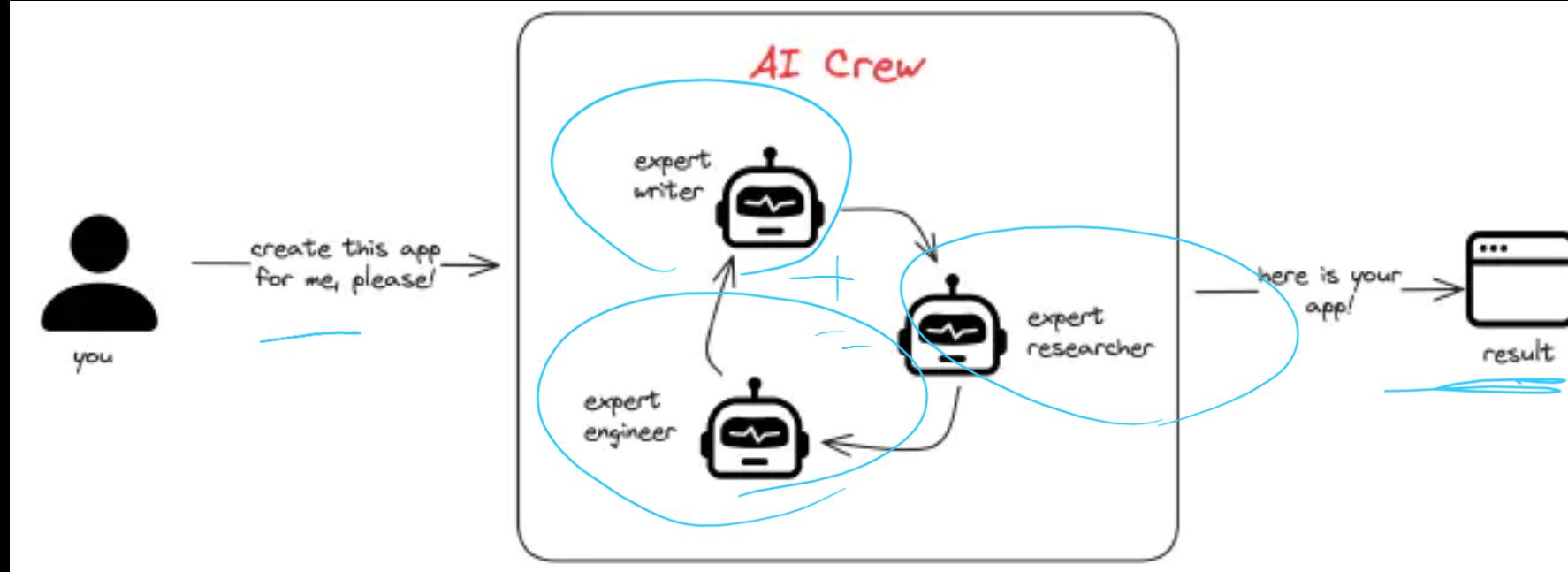
- **Autonomy:** Agents decide their own actions without central control.
- **Interaction:** Agents communicate with each other (messages, shared memory, APIs).
- **Goal-Oriented:** Each agent contributes towards a shared or individual objective.

• Example in AI:

- Research agent gathers information.
- Summarizer agent processes it.
- Publisher agent posts it.



↳ Create tasks
 ↳ description → Expected outputs.
 ↳ You add you mention Agent



Why We Need Multi-Agent Systems



Scalability: Split big, complex tasks into smaller, independent tasks.

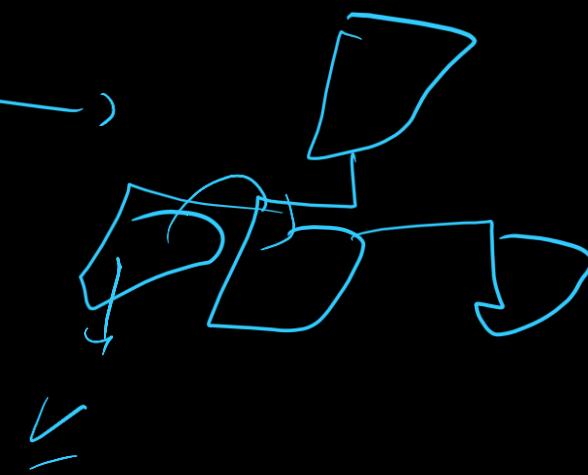
Specialization: Assign tasks to agents best suited for them (e.g., language translation vs. fact-checking).

Parallel Execution: Tasks can run simultaneously → reduced total completion time.

Fault Tolerance: If one agent fails, others can still continue.

Adaptability: Agents can change strategies based on environment feedback.

Example: A content pipeline – Research, Write, Review, Translate, Publish.



Static vs Dynamic Role Assignment

add

✓ Static Role Assignment

Roles are fixed before execution.

Predictable flow and easier to debug.

Good for **repeatable, standard workflows**.

} crew

Example:

Agent A = Researcher

Agent B = Writer

Agent C = Reviewer (always fixed)

✗ Dynamic Role Assignment

Roles assigned **at runtime** based on task context or environment state.

Requires **strong planning logic** or **task routing** mechanism.

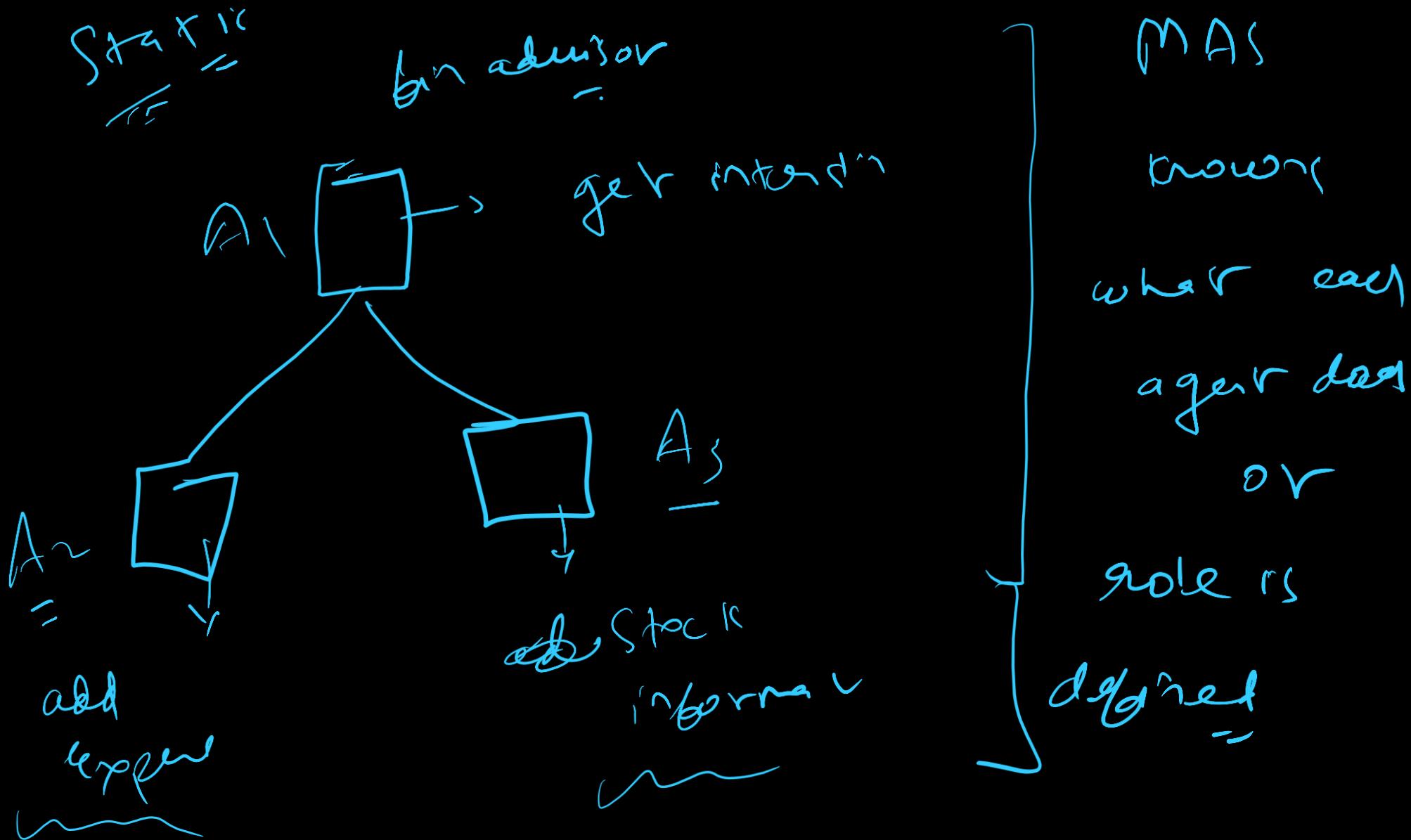
Good for **adaptive workflows** where priorities shift.

Example:

Any available agent can become Reviewer if output quality is low.

fun

long graph



Dynamic

↳ poles are not distinct

↳ you give the agent option to do
whatever it may to achieve the goal
(Please do this in ↗
specificestrich.)

Analogy of a Multi-Agent System

Film Production Crew Analogy

Static example:

Director directs, Scriptwriter writes, Cameraperson films, Editor edits.

Everyone sticks to their defined role.

Dynamic example:

- If the Editor is sick, the Cameraperson temporarily edits scenes.

Lesson: MAS can function like teams in real life — sometimes with fixed responsibilities, sometimes flexible based on the situation.

Real-World Use Cases

Autonomous Vehicles:

Navigation agent, obstacle detection agent, traffic signal reader, decision-making agent.

Financial Trading:

Market data collector, signal analyzer, order executor, risk manager.

Customer Service:

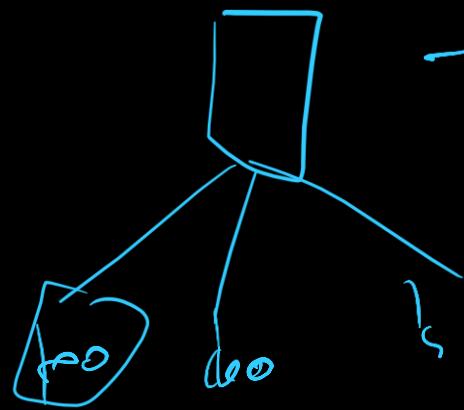
Chatbot for FAQs, escalation bot for complex issues, analytics bot for feedback trends.

Healthcare:

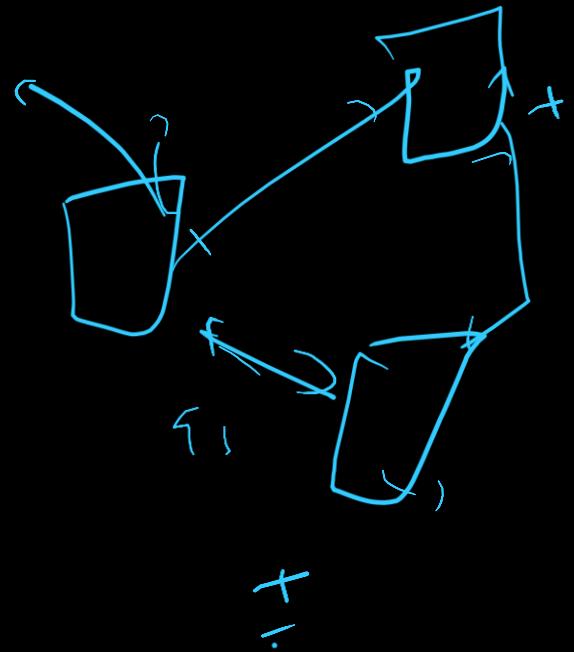
Symptom checker agent, diagnosis agent, treatment planner, patient follow-up bot.

Content Creation Pipelines:

Topic researcher, writer, fact-checker, translator, publisher.

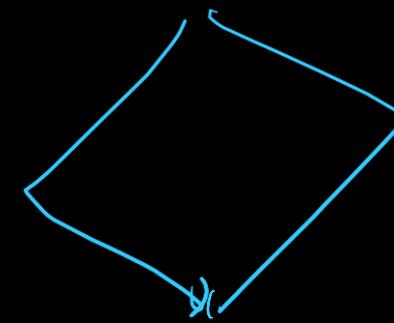


→ Dynam.



↳ ib - elv
stan

↳ Autonomy



Types of Multi-Agent Systems



Network-based

Agents are connected in a network and communicate peer-to-peer.
No single leader; coordination happens through shared messages.
Good for collaborative brainstorming and distributed decision-making.

Supervisor-based

One main agent manages and directs other agents.
Centralized decision-making; sub-agents follow instructions.
Works well when one agent has full context of the goal.

Supervisor + Tool Calling

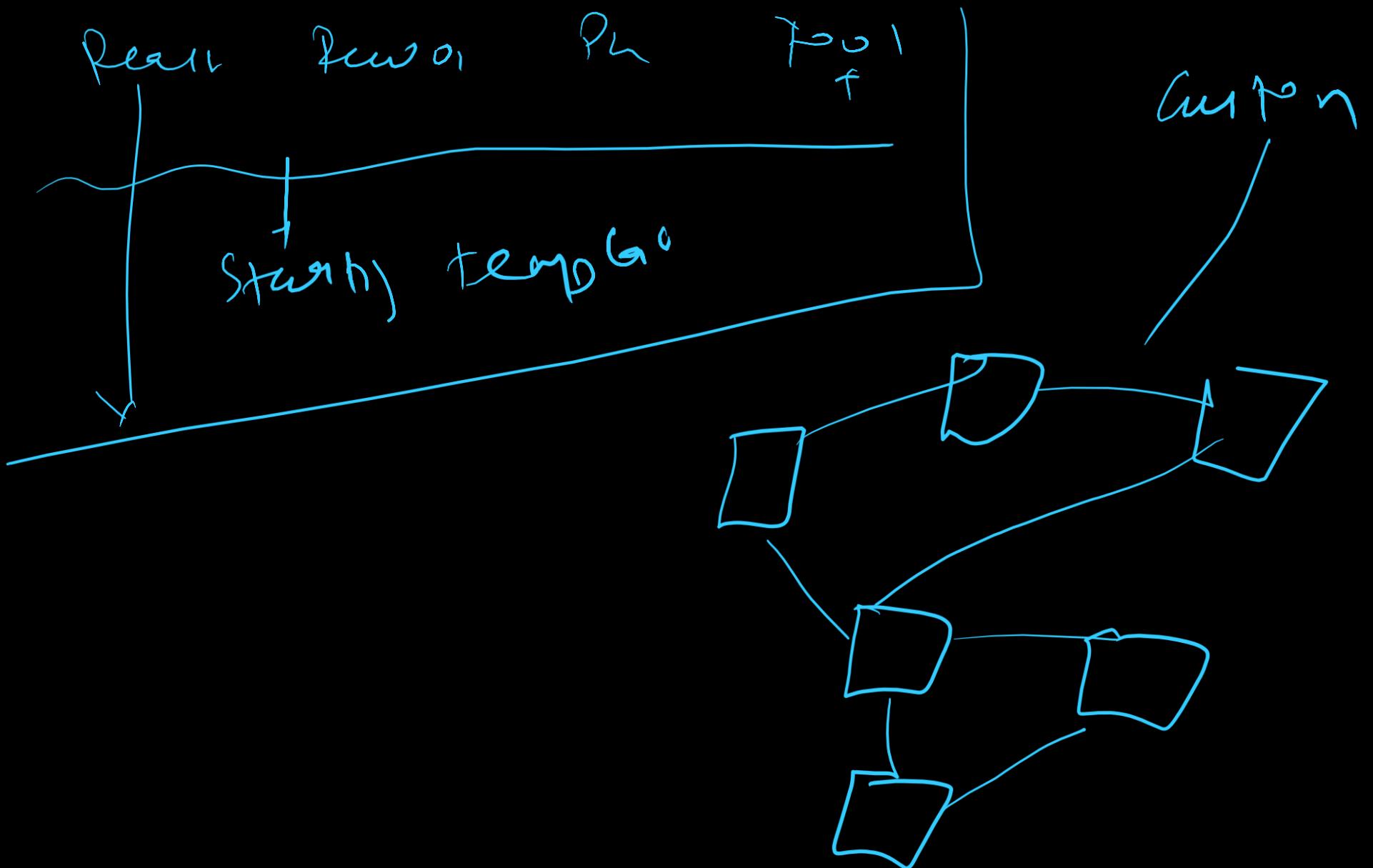
Supervisor agent delegates tasks and decides when to call external APIs/tools.
Ideal for workflows that mix reasoning with real-world actions.
Adds automation without losing centralized control.

Hierarchical

Agents are organized in layers: each layer supervises the one below.
Breaks large goals into smaller sub-goals at different levels.
Effective for multi-step complex projects with dependencies.

Custom Multi-Agent Workflows

Fully tailored to the problem; roles, flows, and rules are custom-coded.
May mix network, supervisor, and hierarchical patterns.
Offers maximum flexibility but requires more design effort.

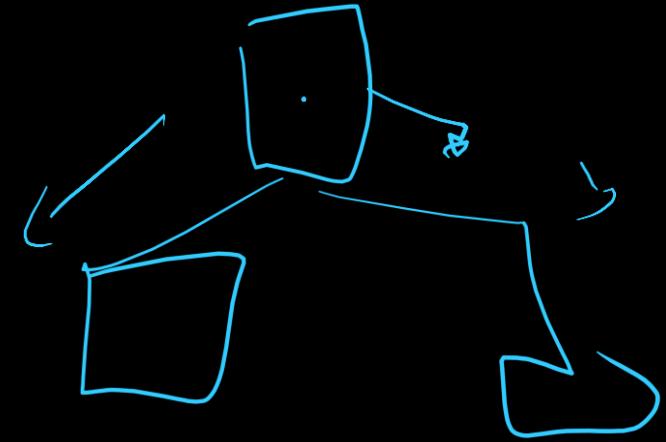
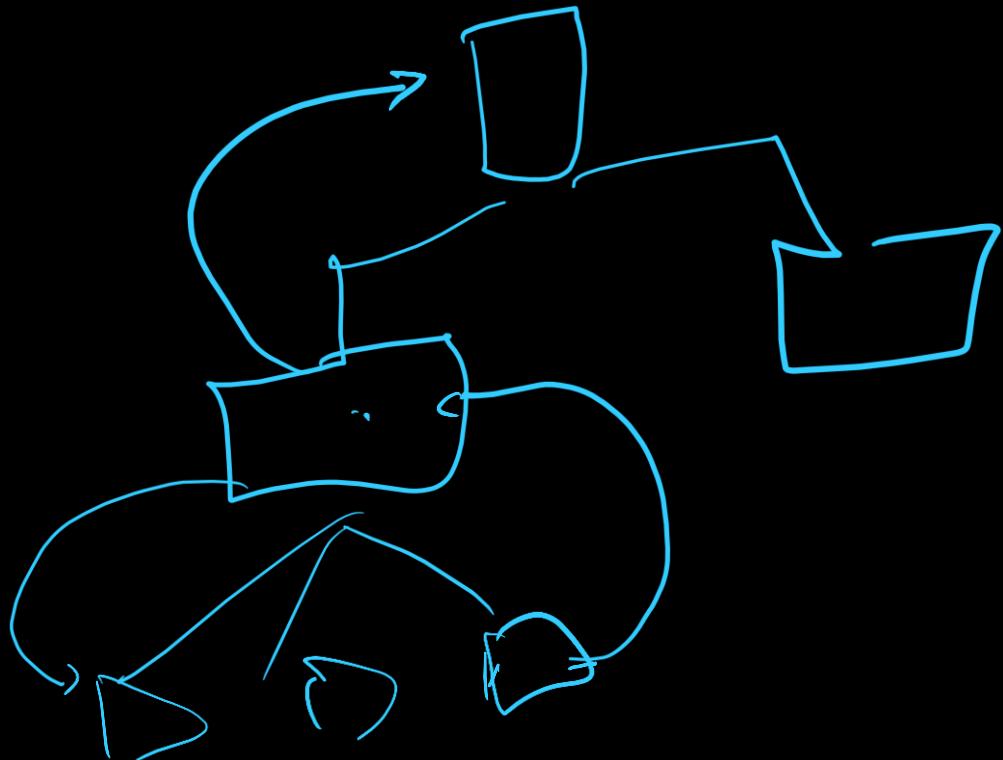


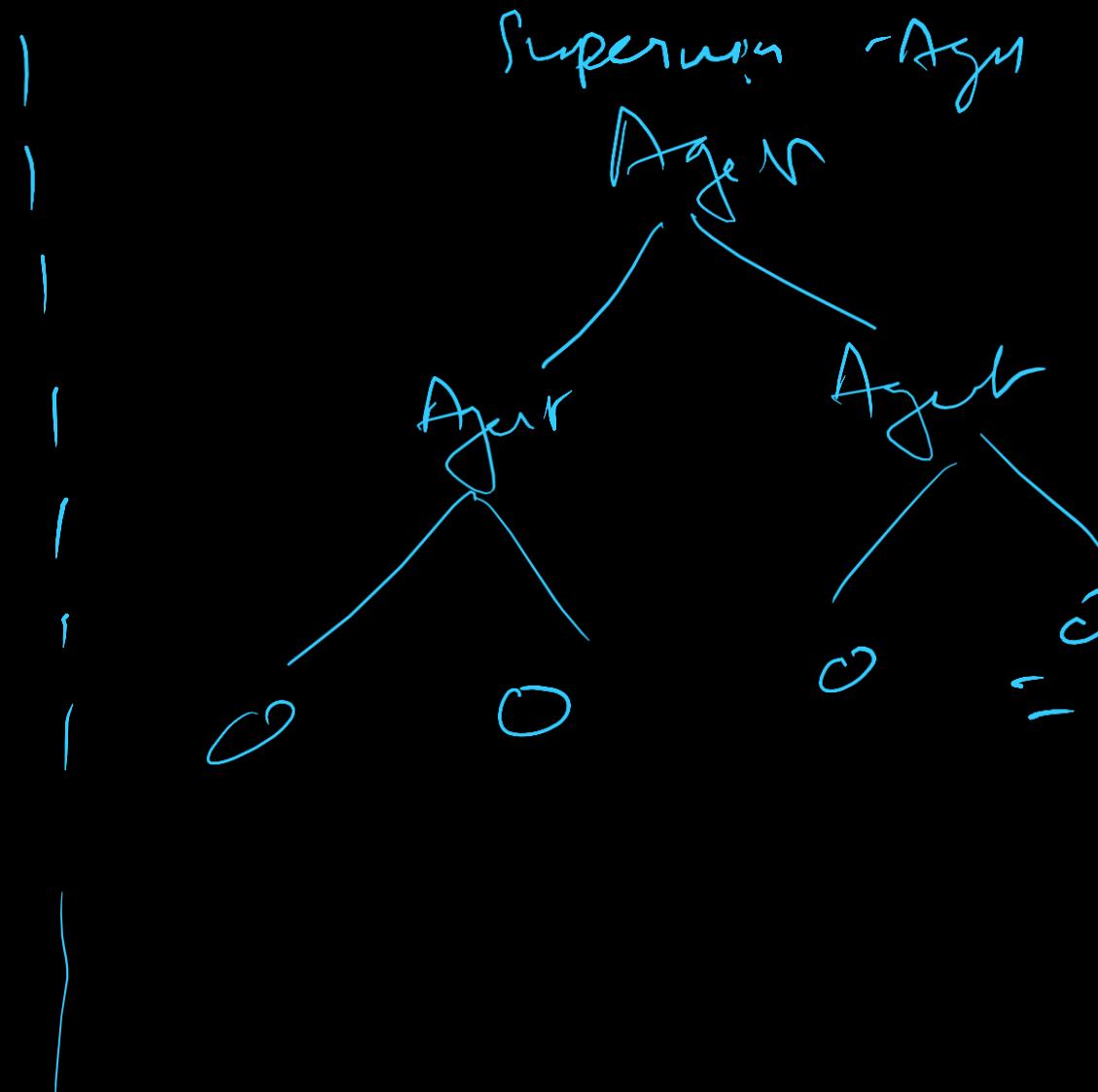
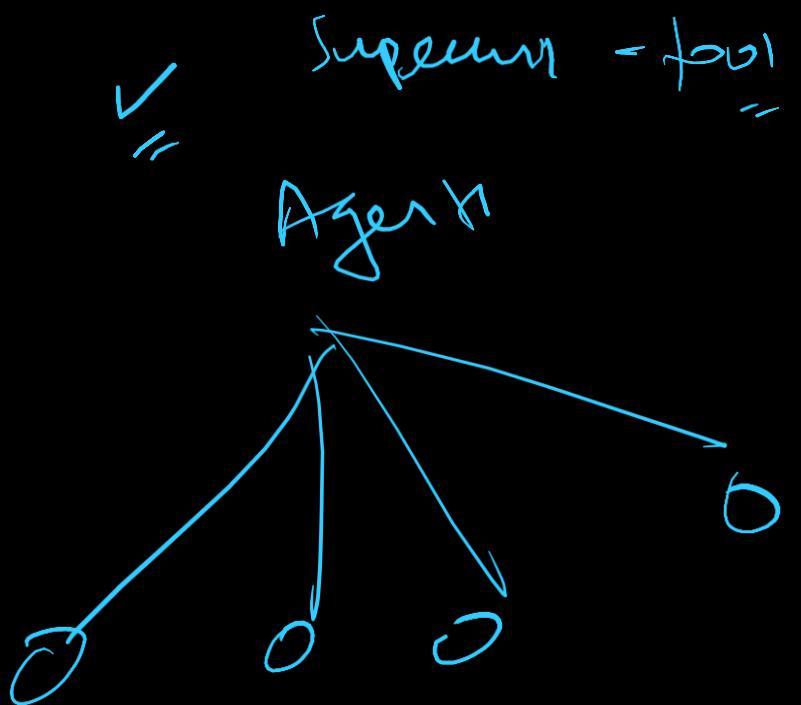
Kierch's

=

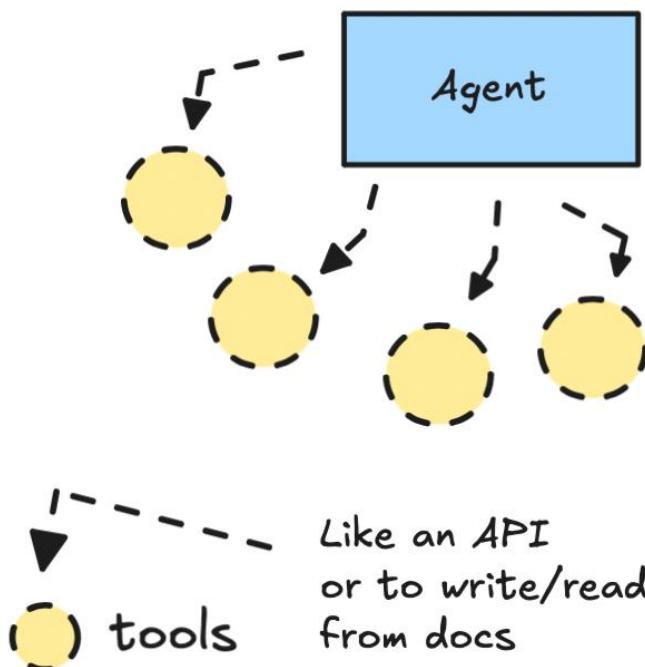
micronay

lomp

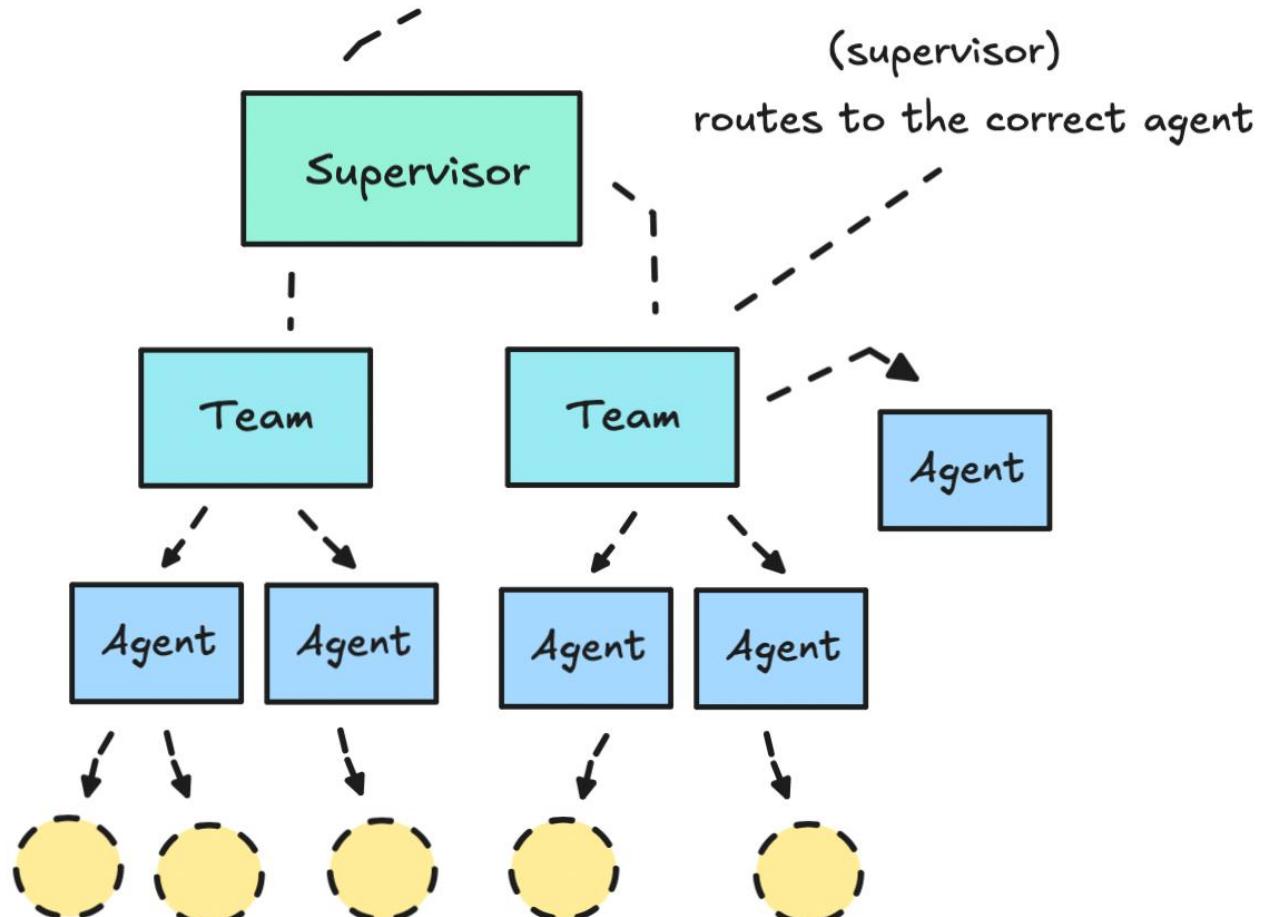


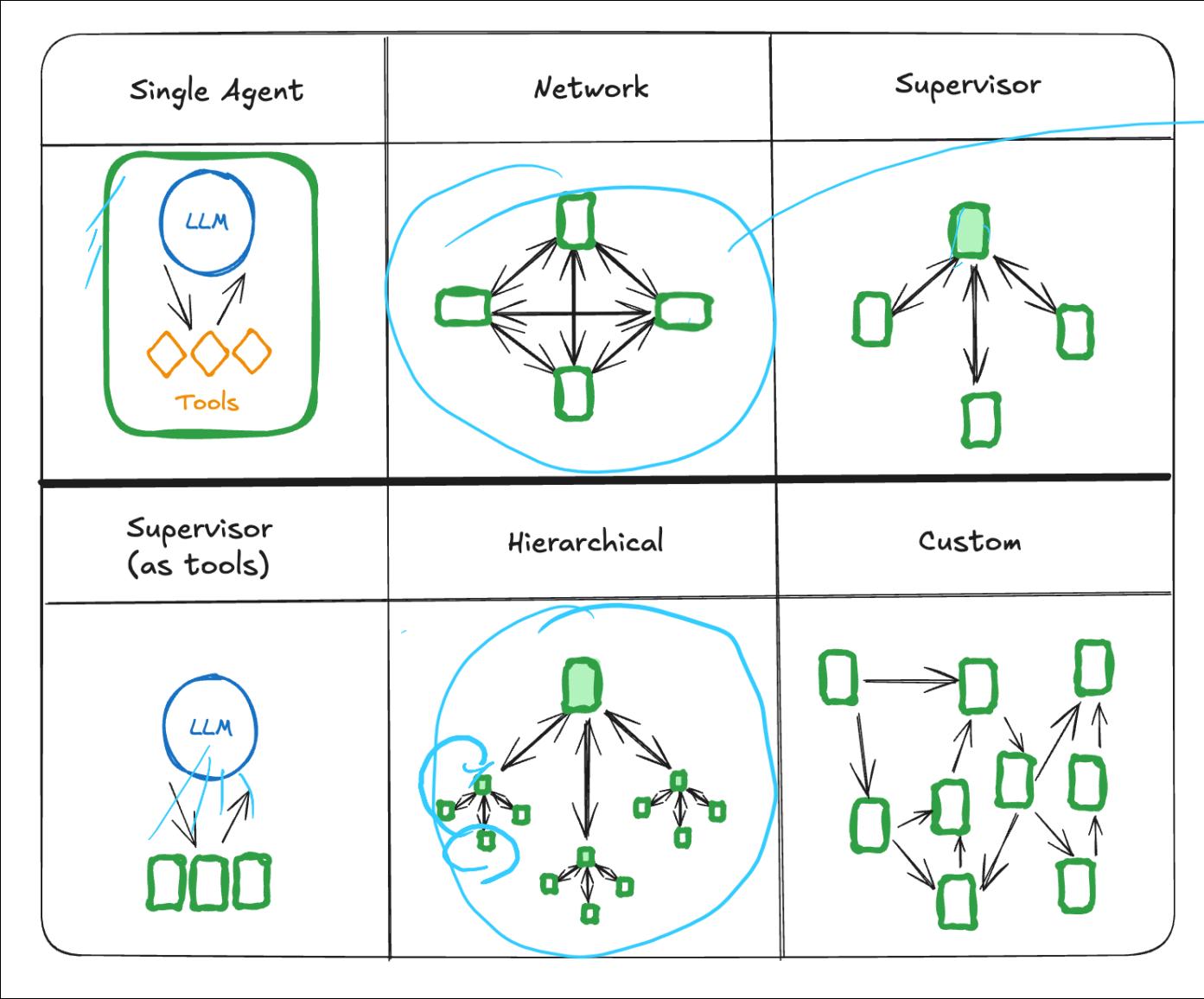


Single vs Multiagent Systems



routes to the correct team

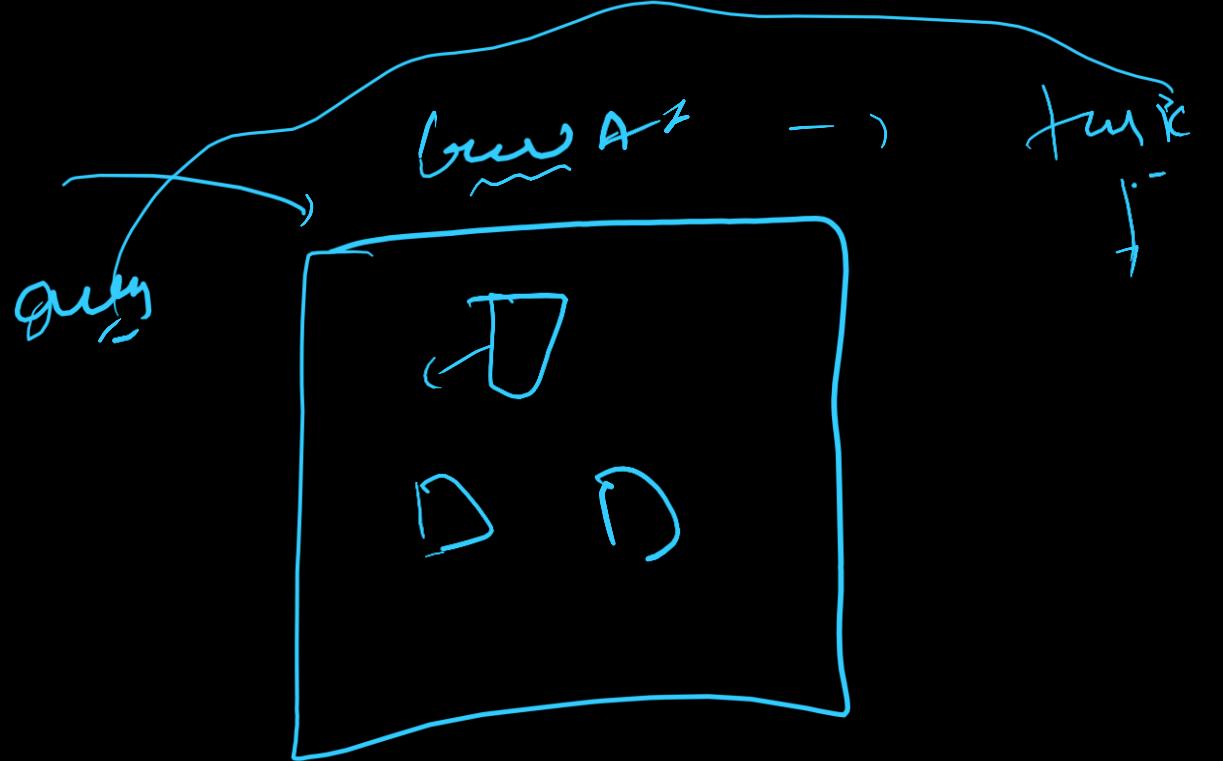




Loggraph -> graph

Graph \rightarrow $\vdash \exists \rightarrow$ autom build for multi Agent
System

Autogen in



Feature

CrewAI

LangGraph

Approach

Role-based task orchestration

Graph-based workflow
orchestration

❖ Best for

Predefined, static workflows

Adaptive, dynamic workflows

Ease of setup

High – simple config

Moderate – requires node/edge
setup

Parallelism

Supported

Supported with branching

Control over flow

Linear/sequential/parallel tasks

Complex decision-based paths

Dynamic Role Switching

Limited

Strong support

Multi-Agent System with CrewAI

- **How it works:**

- Define agents → assign roles, goals, and tools.
- Define tasks → link them in sequence or parallel.
- CrewAI orchestrates task execution based on role definitions.

- **Strengths:**

- Simple syntax for agent definition.
- Quick setup for pipelines with fixed stages.

- **Example Workflow:**

- Agent 1 (Researcher) → Agent 2 (Writer) → Agent 3 (Reviewer) → Agent 4 (Publisher).

Term	Meaning	Example
Role	The function or job of the agent in the crew.	"Content Writer"
Goal	What the agent is trying to achieve.	"Write an engaging blog post on AI trends."
Backstory	Context or persona that guides the agent's behavior & style. <i>context</i>	"An experienced tech journalist with a witty tone."
Other Terms		
<ul style="list-style-type: none">• verbose=True → Agent outputs detailed reasoning & logs during execution (good for debugging).• allow_delegation=True → Agent can assign tasks to other agents in the crew if needed.		

Real-World Agent Team: Planner, Executor, Critic

Planner:

Breaks high-level goal into specific tasks.
Assigns tasks to the right agents.

Executor:

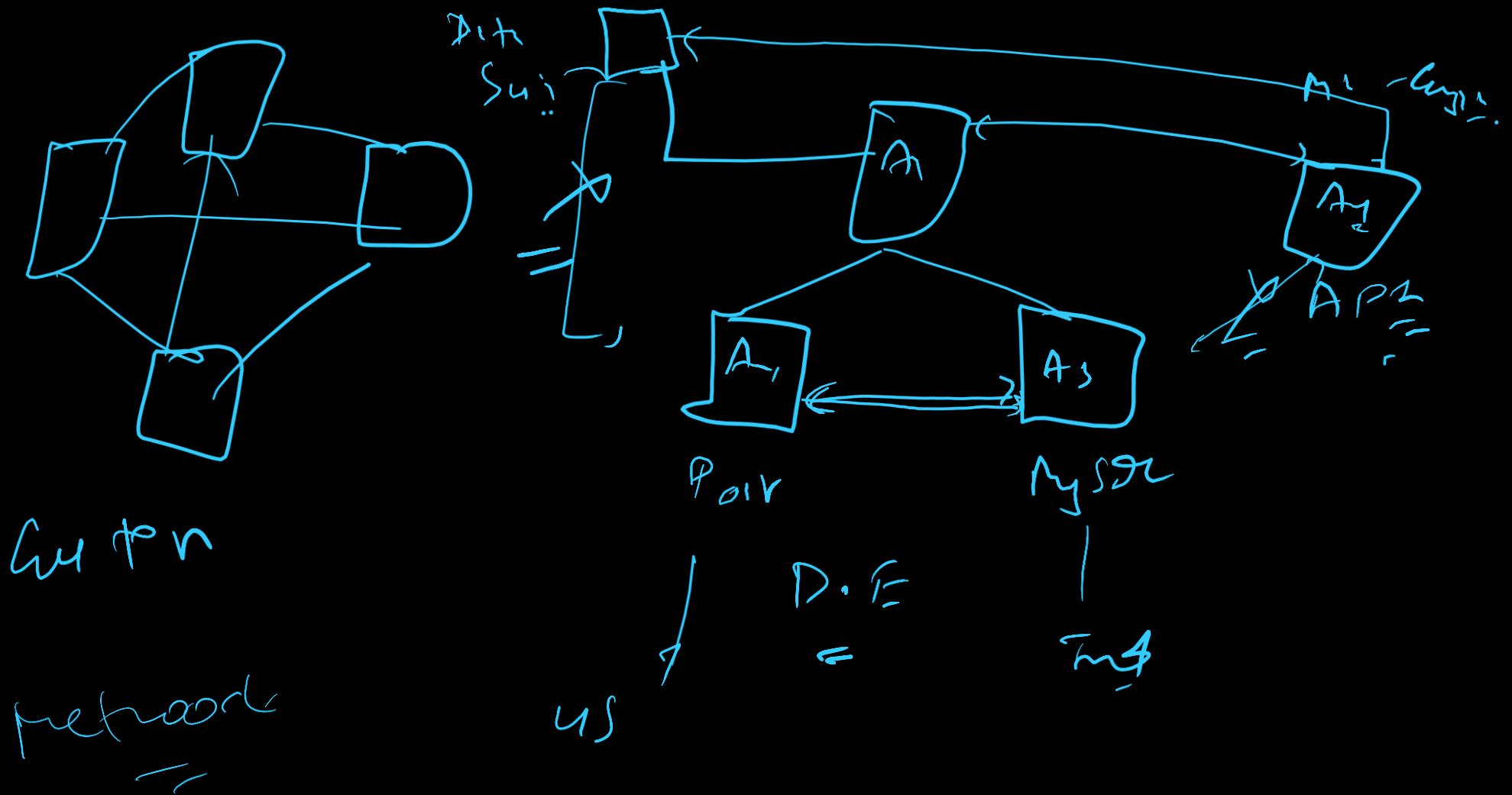
Performs assigned task using tools, APIs, or models.
Returns results to the Planner or Critic.

Critic:

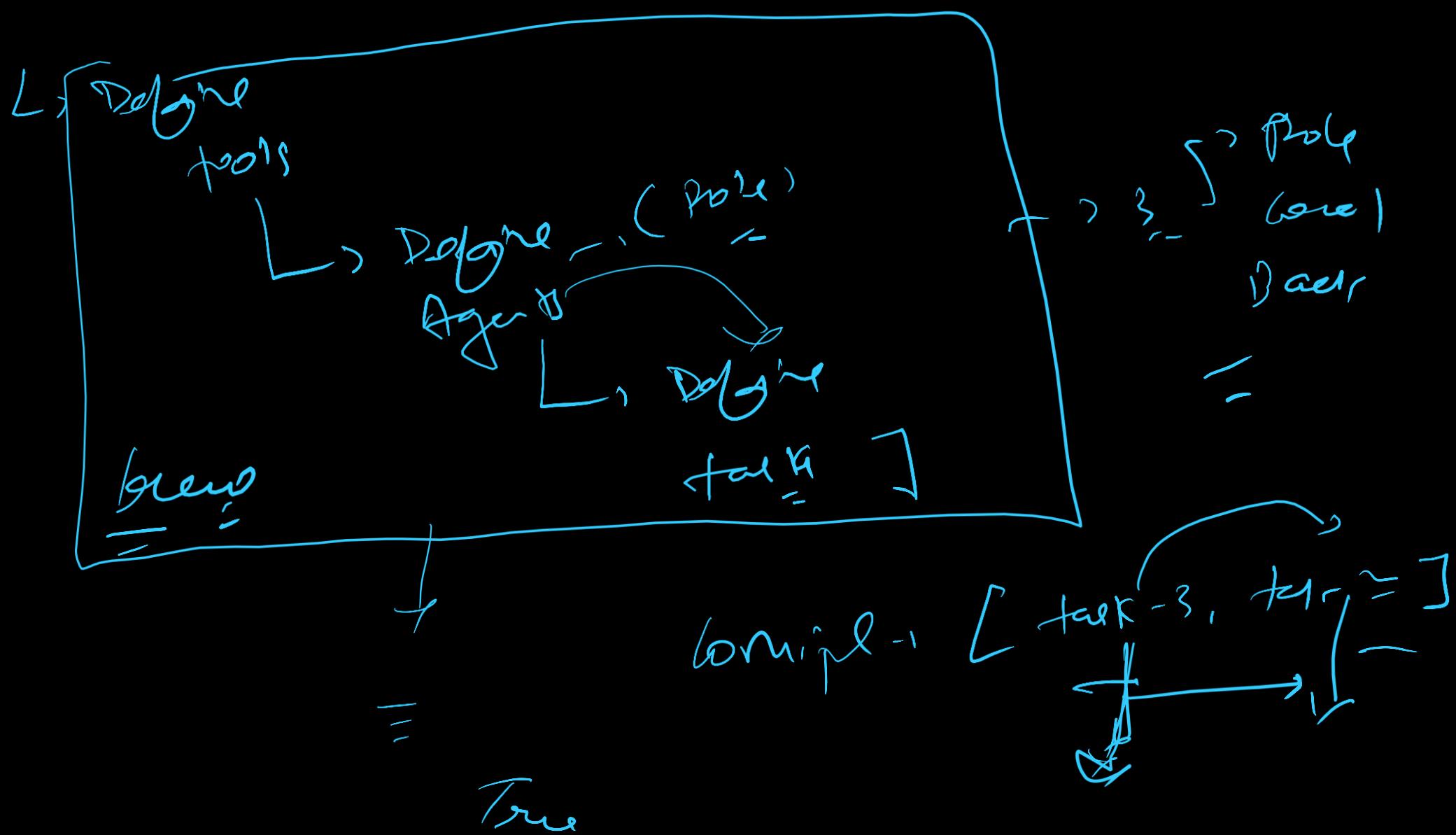
Evaluates output for quality, accuracy, and completeness.
Suggests corrections or improvements before finalizing.

Loop:

Planner → Executor → Critic → Adjust → Final Output.



- ↳ static role based inferred goals
 - ↳ goal (\rightarrow)
 - ↳ role \rightarrow TD
 - ↳ Back step: context
 $\underline{E_{\text{out}}^{\text{in}}} = (\text{And},$
-), iH make generic
Systm look Simpl.



Dear
J =

Naught : "



"