OPEN AI'S EXPERIMENTAL FRAMEWORK

What is SWARM?

- The Swarm Framework is an open-source tool developed by OpenAI to help create, manage, and coordinate multiple AI agents that work together like a team.
- Think of it as a way to organize smart AI bots to handle complex tasks by dividing the work among them.

Early Days (2023–2024)

- ***Idea and Creation:*** Swarm was created by OpenAI in 2023 as an experimental tool to let **multiple AI agents work together**, like a team or a swarm of bees.
- ***Purpose:*** Instead of one big AI doing everything, Swarm let you build **specialized agents**—each doing a specific job and sharing info to solve complex tasks.
- ***Inspiration:*** It was inspired by **multi-agent systems** in AI research, where teams of agents (like animals or people) **collaborate to reach a goal**.

SWARM

Key Features

- ***Lightweight and Flexible:*** Built in Python, Swarm made it easy to create and customize AI agents with specific roles and rules.
- ***Agent Coordination:*** Agents could **communicate**, share tasks, and use external tools like APIs or databases.
- ***Open-Source:*** Swarm was free and available on GitHub, so anyone could use it, experiment, or improve it.

SWARM

Evolution (2024–2025)

- ***Growing Popularity:*** By 2024, Swarm was used for things like customer support, simulations, and app workflows—thanks to its flexibility.
- ***Improvements:*** OpenAI and the community improved agent communication, added support for complex tasks, and made it work better with models like ChatGPT.
- ***Community Contributions:*** Developers worldwide contributed code, tutorials, and ideas, helping Swarm evolve quickly.

SWARM

Why It Matters

- Swarm made it easy to build **teams of AI agents**, each with a specific role—like a virtual call center or a group of assistants working together.
- One agent could do math, another could write, and they'd **collaborate smoothly** to solve bigger problems.

Current Status (2025)

- Swarm is still **experimental**, but actively developed by OpenAI and the community.
- It's mainly used for **prototyping and research**, and while not widely known, it's a **powerful tool for building multi-agent AI systems**.

 Swarm Framework (Experimental)

Swarm was OpenAI's **lightweight tool** for building multi-agent AI systems. It introduced two key ideas:

- **Agents**: Small, specialized AIs, each with its own tools and tasks.
- **Handoffs**: A way for agents to **pass tasks** between each other based on context (like routing a billing question from a general bot to a billing agent).

Swarm focused on **simplicity and flexibility**, helping developers experiment with how AIs could collaborate like a well-coordinated team.

🚀 OPEN AI Agents SDK (Production-Ready)

In recent updates, OpenAI released the **Agents SDK**, a **more powerful and production-ready evolution of Swarm**.

- It builds on Swarm's core ideas—**autonomous agents and collaboration**.
- Adds **better tools, structure, and coordination features** for real-world applications.
- Makes it easier to manage complex workflows with multiple agents working together.

Why It Matters

- Swarm laid the groundwork for building **collaborative AI systems**, and the Agents SDK turns those ideas into something you can use in **real apps and services**—from customer support to automated research teams.

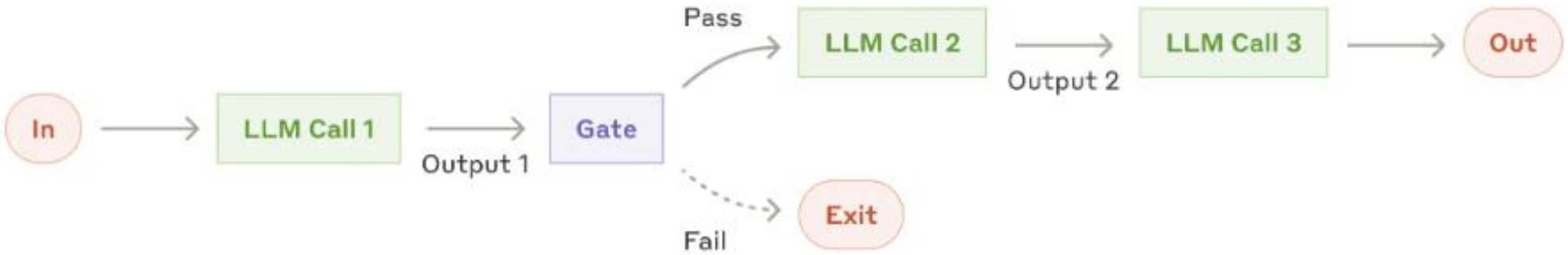
Anthropic Design Patterns

- **OpenAI's Agents SDK** is a flexible tool that helps developers build and manage AI agents that can handle complex tasks.
- It supports several proven **design patterns** (like those from **Anthropic**) to make agent collaboration smoother and more effective.
- Developers can easily use these patterns to create smarter and more organized AI systems.

1. Prompt Chaining (Chain Workflow)

- **What it means:** Break a big task into smaller, simple steps done one after another.
- **How it works with Agents SDK:**
  - You can set up agents to handle each step in a specific order, passing results from one to the next.
  - Each LLM call processes the output of the previous one. You can add programmatic checks (see "gate" in the diagram) on any intermediate steps to ensure that the process is still on track.
- **Why it helps:** Keeps the process organized and easier to manage, especially for complex tasks.
- **Example:** To write a research summary:
  - One agent **searches for sources**, Another **extracts key points**, A third **writes the summary**.

1. Prompt Chaining (Chain Workflow)



The prompt chaining workflow

2. Routing

What it means:

- Send each task to the right agent based on what the task is about.

How it works with Agents SDK:

- Agents can handoff tasks to other agents that are better suited for specific jobs (e.g., billing, tech support).

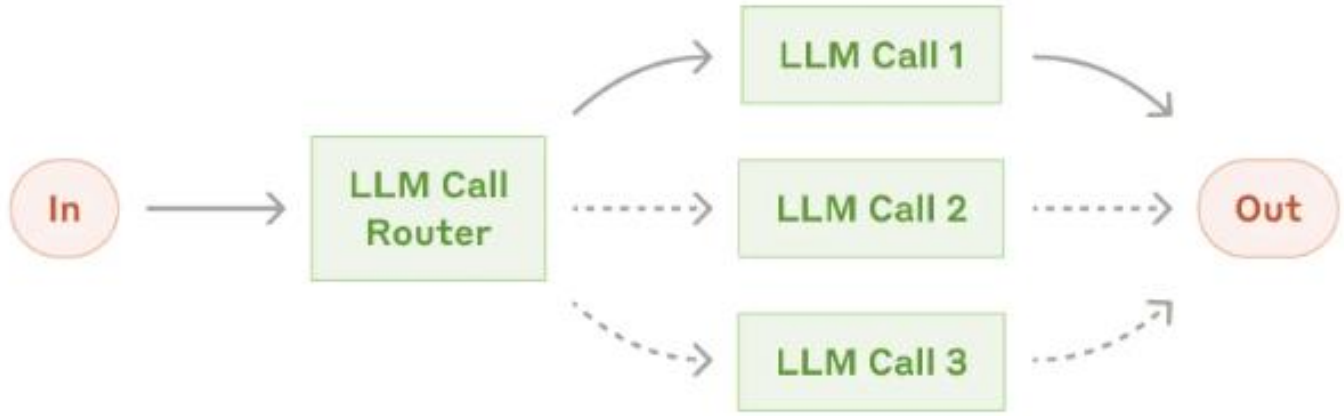
Why it helps:

- Makes sure each part of the task is handled by the most capable agent, improving efficiency and accuracy.

Example:

- A general support agent receives a question about a payment issue → it **routes** the task to a **billing agent** who handles it.

2. Routing

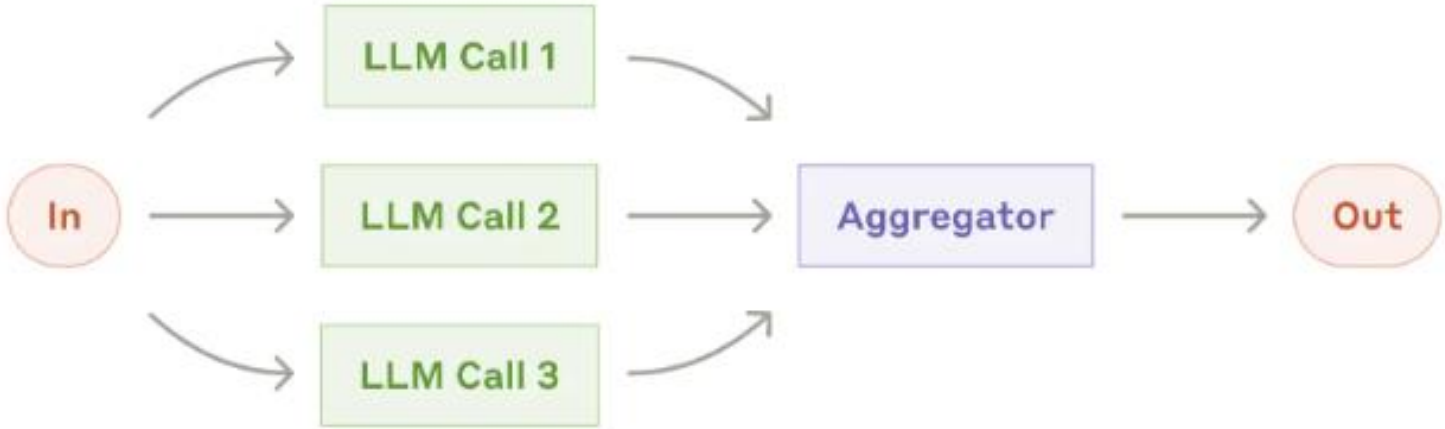


The routing workflow

3. Parallelization

- **What it means:** Run multiple subtasks at the same time instead of one after another.
- **How it works with Agents SDK:** You can set up agents to work in parallel, each handling a different part of the job at once.
- **Why it helps:** Saves time and increases overall system efficiency.
- **Example:** For a market report:
  - One agent gathers pricing data,
  - Another analyzes trends,
  - A third summarizes insights — all at the same time.

3. Parallelization



The parallelization workflow

4. Orchestrator–Workers

- **What it means:** One main agent (orchestrator) breaks down a task and gives parts of it to helper agents (workers).
- **How it works with Agents SDK:** The orchestrator manages the flow and assigns each subtask to the right worker agent.
- **Why it helps:** Keeps large tasks organized and makes teamwork between agents more efficient.
- **Example:** For building a website:
  - The orchestrator plans the project,
  - One worker writes the HTML,
  - Another handles the design,
  - Another sets up the backend.

4. Orchestrator–Workers

The orchestrator-workers workflow

5. Evaluator–Optimizer

- **What it means:** One agent reviews the work of other agents and suggests improvements.
- **How it works with Agents SDK:** Using guardrails, an evaluator agent can check outputs, give feedback, and help optimize results.
- **Why it helps:** Ensures better quality, fewer mistakes, and continuous improvement.
- **Example:** After a writing agent creates a blog post, an evaluator agent checks for grammar, clarity, and tone, then suggests edits for improvement.

5. Evaluator–Optimizer

The evaluator-optimizer workflow