Open AI's Swarm:

Swarm Open AI ka ek experimental, lightweight aur opensource framework hai jise multi-agents system banane k liye design kiya gaya hai. Yeh system flexible, simple aur scalable hai. Multi agents mil kar complex kaam asani se kar sakte hain.

OpenAI Swarm framework mai "Agents" aur "Handoffs" 2 fundamental abstractions hain jo multi agent coordination ko asaan banate hain.

What is Agents abstraction?

Har agent ek encapsulated unit hai, jis mai ap instructions, model aur tool define karte hain. Apko pata nahi hota k LLM internally query ko kaise tokenizes karta hai, bus ap instructions dete ho aur agent output deta hai.

Agents & Handoffs:

Agents:

Agents ek tarah k intelligent components hote hain jo large language model(LLM) pe dependant hote hain. Yeh instructions aur tools k zariye tasks ko execute karte hain. Har agent apna kaam alag karta hai. Har agent k pas apni instructions(system prompts) aur tools(python functions) hote hain.

Example: English Agent, Spanish Agent, har agent apne role mai specialized hota hai.

Handoffs:

Handoffs wo sub-agents hote hain jinko ek main agent task delegate karta hai, take har task ziada appropriate specialized k pas jaye. Jab ek

agent dusre agent ko kaam handover karta hai to isay “Handoff” kehte hain.

Example: Triage Agent pehle user ka input access karta hai, phir appropriate agent ko forward karta hai jaise billing, refund ya sales.

Open AI Agents SDK:

Open AI ne recently Agents SDK release kiya hai jo k Swarm ka production-ready, upgraded version hai. Yeh SDK Swarm k basic concepts jaise Agents aur Handoffs ko ziada advanced features aur robustness k sath real world use cases k sath fit karta hai. Isse developers complex workflows easily organize kar sakte hain, jahan multiple agents coordinated tor par ek dusre k sath mil kar kaam karte hain, take sub agents mil kar better result perform kar saken.

Anthropic Design Patterns (Jo SDK Follow karta hai):

OpenAI ka Agents SDK aik versatile (bohat flexible aur adaptable) framework hai, jo AI agents ke development aur orchestration (munazzam taur par integration) ko asaan banata hai, is framework ki madad se agents mushkil tasks efficient taur par kar sakte hain. Yeh SDK Anthropic ke propose kiye gaye design patterns ke saath kaafi closely aligned hai, jisse developers in patterns ko bina kisi mushkil ke seamlessly implement kar sakte hain."

Prompt Chaining:

Task ko chhote chhote steps mein taqseem karna — har agent apna step karta hai.
Example: Tumhe kitab ka ek hissa summarize karna hai:

1. Agent-1 (Extraction agent): “Sab se important quotes nikal do.”
2. Agent-2 (Summary agent): Un quotes ka concise summary banao.
3. Agent-3 (Refinement agent): Summary ko polish karo, readability achi banao.
   Har step ka output agla step ka input banta hai.

Routing:

Task ko sab se suitable agent ke paas bhejna.
Example: Customer service chatbot:

- Router agent pehle message ko analyze karta hai.
- Agar user “billing issue” kahe → Billing Agent ko bhejta hai.
- Agar “technical error” ho → Tech Support Agent ko forward karta hai.
  Is se har sawal ko us agent ke pass bheja jata hai jo us kaam mein mahir ho .

⚡ Parallelization:

Multiple agents ek saath kaam karte hain taake speed barhe.
Example: Tumhare paas ek bada text hai aur summary chahiye:

- Agent A themes extract karta hai.
- Agent B sentiment analyze karta hai.

- Agent C factual inconsistencies check karta hai.
  Teenon ek saath chal kar apna kaam karte hain, phir result aggregate hota hai.

👥 Orchestrator-Workers:

Ek main agent (orchestrator) task torhta hai aur worker agents ko assign karta hai.
Example: Trip itinerary banana hai:

- Orchestrator agent decide karta hai ke kaunse cities visit karne hain.
  - Phir Destination Agent decide karta hai cities.
  - Itinerary Agent din war plan banata hai.
  - Booking Agent hotels/flights arrange karta hai.
    Orchestrator sab results ikattha karke final plan banata hai.

Evaluator-Optimizer:

Ek agent doosre ke output ko assess karta hai aur improvement suggestions deta hai.
Example: Text ko translate karna hai:

1. Translator Agent translation deta hai.
2. Evaluator Agent accuracy aur tone check karta hai.
3. Optimizer Agent translation improve karta hai.
   Ye loop chalta rahega until final quality achieve ho jaye.

✅ Kyun zaroori hain ye patterns?

- **Accuracy aur transparency:** Agar step mein problem ho, sirf usi step ko theek karo.
- **Efficiency:** Multiple agents ek saath kaam kar ke time bachate hain.
- **Specialization:** Har agent apne domain mein expert hota hai.

created by Zoya Afzal