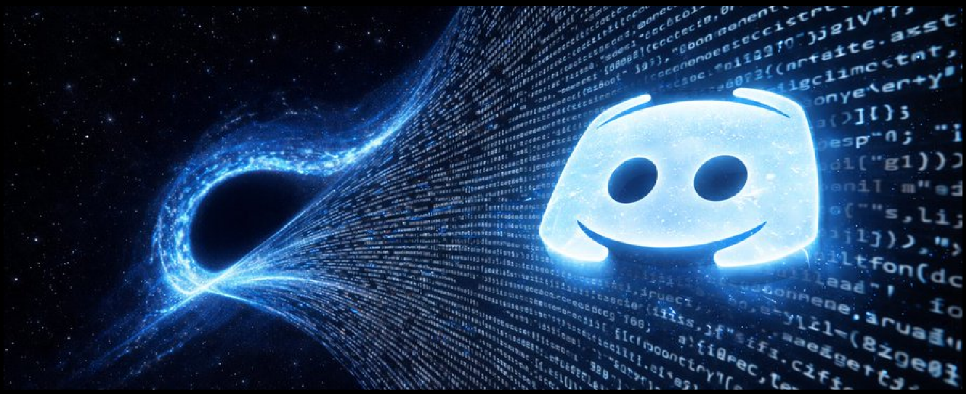


I Built an AI Agent Swarm in Discord. It Works Better Than Anything I’ve Tried (Full Guide)

JUMPERZ
@jumperz · Feb 8

Follow

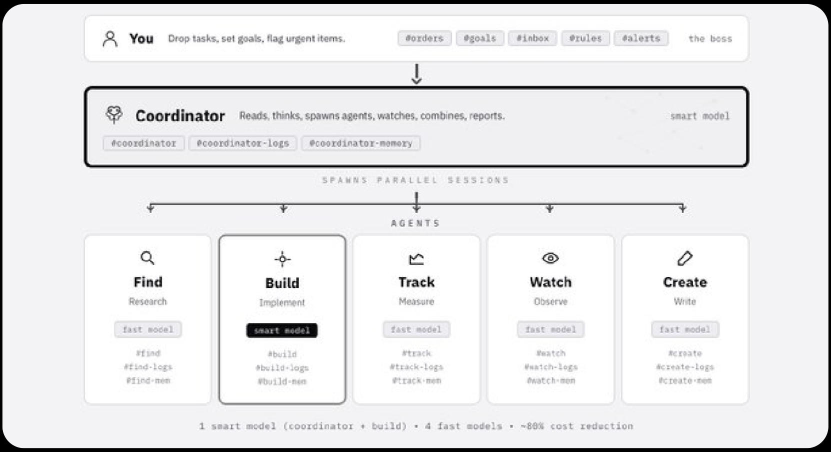
41 63 720 83K

So I built this agent coordination system that lives in my Discord server.

they talk to each other, split work, and deliver results, I was surprised how easily this actually worked out. It's honestly kinda wild.

let me walk you through everything..

# The Entry Point (YOU)



You type plain English in #orders channel.. no special commands. its a discord channel where everything starts.

- "Research the top 10 AI coding tools and write a comparison thread"
- "Build a landing page for my new SaaS idea"
- "Track engagement on my last 5 tweets and suggest improvements"

That's it, just natural language..just drop your task like you're texting a friend.

=========

# The Brain (Coordinator)

One smart model reads your task, thinks about what needs to happen, breaks it into pieces, spawns the right agents. It never does the work itself. It only thinks and delegates.

Like a manager who actually manages.

The coordinator sees "research AI coding tools" and thinks:

- '**Find** agent': scrape websites, read docs, collect data
- '**Create** agent': write the comparison thread
- '**Track** agent': measure which tools get mentioned most

Then it spawns them all at once.

=========

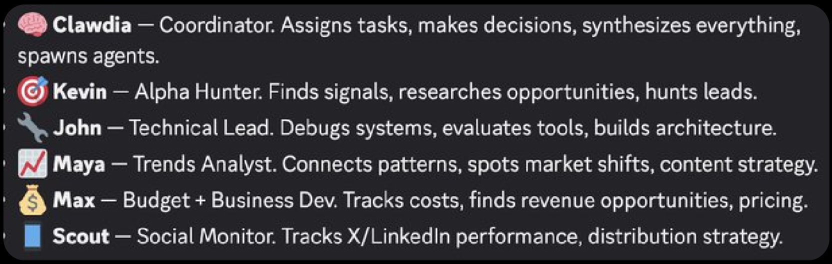
# The Team (5 Agents)

- **Find** (Research): Web scraping, data collection, trend analysis
- **Build**(Implement): Code, websites, automations, integrations
- **Track** (Measure): Analytics, monitoring, performance data
- **Watch** (Observe): Social listening, competitor tracking, alerts
- **Create** (Write): Content, copy, documentation, threads

All run as real independent sessions in parallel. Not one agent pretending to be many. Real separate sessions working at the same time.

PS.
you have to give them names so you easily can remember them later.. mine are called Scout, Max, John, Maya.. etc. so when you need one you call them by name easily and your coordinator which also need a name would remember and recognises them way easier.

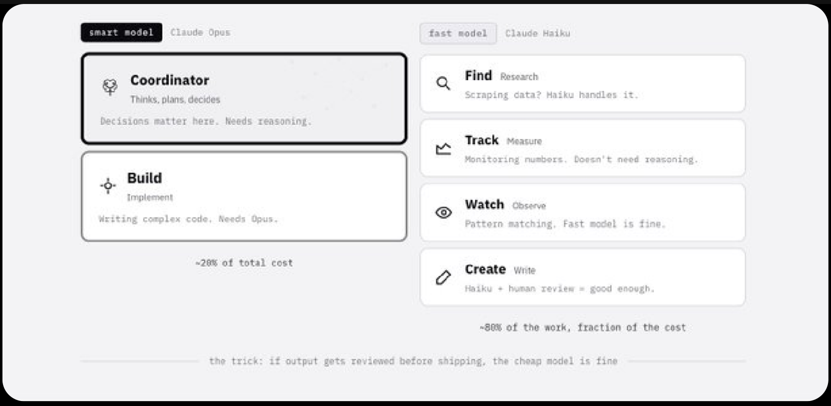
Example:



🧠 **Clawdia** — Coordinator. Assigns tasks, makes decisions, synthesizes everything, spawns agents.
🎯 **Kevin** — Alpha Hunter. Finds signals, researches opportunities, hunts leads.
🔧 **John** — Technical Lead. Debugs systems, evaluates tools, builds architecture.
📈 **Maya** — Trends Analyst. Connects patterns, spots market shifts, content strategy.
💰 **Max** — Budget + Business Dev. Tracks costs, finds revenue opportunities, pricing.
📘 **Scout** — Social Monitor. Tracks X/LinkedIn performance, distribution strategy.

=========

# The Model Tier Trick


smart model
Claude Opus
Coordinator
Thinks, plans, decides
Decisions matter here. Needs reasoning.
Build
Implement
Writing complex code. Needs Opus.
~20% of total cost
fast model
Claude Haiku
Find Research
Scraping data? Haiku handles it.
Track Measure
Monitoring numbers. Doesn't need reasoning.
Watch Observe
Pattern matching. Fast model is fine.
Create Write
Haiku + human review = good enough.
~80% of the work, fraction of the cost
the trick: if output gets reviewed before shipping, the cheap model is fine

only the coordinator and builder use the expensive smart model (Claude Opus).. everything else runs on the fast cheap one (Claude Haiku. Sonnet) whatever you want, same quality and 80% less cost.

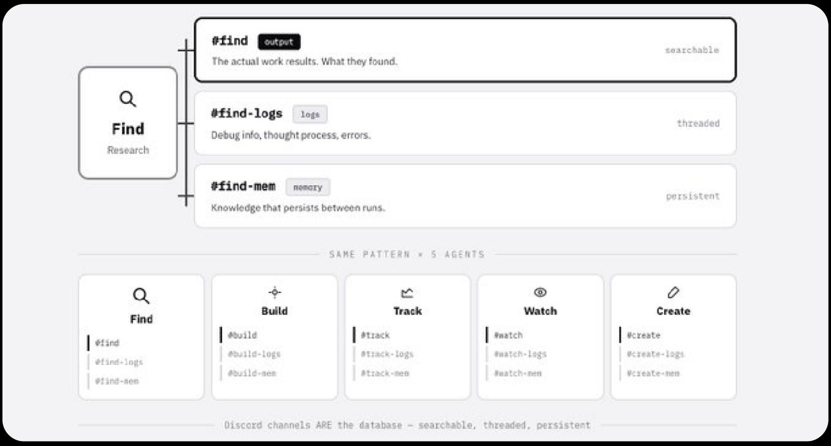
if the output gets reviewed before shipping, the cheap model is fine because you're not publishing raw AI output anyway.

- Find agent scraping data? Haiku can do that.
- Create agent writing a thread? Sonnet + human review = good enough.
- Build agent writing complex code? That needs Opus.

and so on.. smart routing = budget control.

=========

# Three Channels Per Agent



- **Output** (what they found): The actual work results
- **Logs**(how they did it): Debug info, thought process, errors
- **Memory** (what they learned): Knowledge that persists between runs

Discord channels ARE the database, no postgres, no vector stores, already searchable, threaded, and persistent.

> Want to see what Find agent discovered about AI tools last month?
> Search #find-output.
>
> Need to debug why Build agent failed?
> Check #build-logs.

everything is already organized and you didn't build a single database schema.
=========

# Interns *(you need channels for this)*

Any agent can spawn temporary workers. One job, then gone.

need to analyze 10 articles?

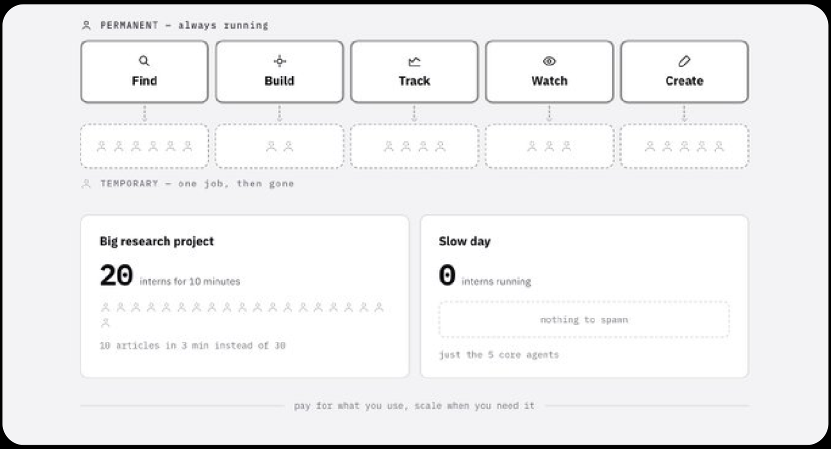
spawn 10 interns, all work at the same time, results in 3 minutes instead of 30.

that's how it scales without permanent cost.

the 5 main agents are always available but the interns come and go based on workload.

> Big research project = 20 interns for 10 minutes.
> Slow day = zero interns.

you basically pay for what you use and scale when you need it.


PERMANENT – always running
Find
Build
Track
Watch
Create
TEMPORARY – one job, then gone
Big research project
20 interns for 10 minutes
10 articles in 3 min instead of 30
Slow day
0 interns running
nothing to spawn
just the 5 core agents
pay for what you use, scale when you need it

=========

# Coordination (#agent-chat)

Agents report "done", "stuck", or hand off to the next agent.

Find hands to Build. Build hands to Track. Like a relay race.

The coordinator watches everything but the agents talk to each other directly... faster than going through the middle man every time.

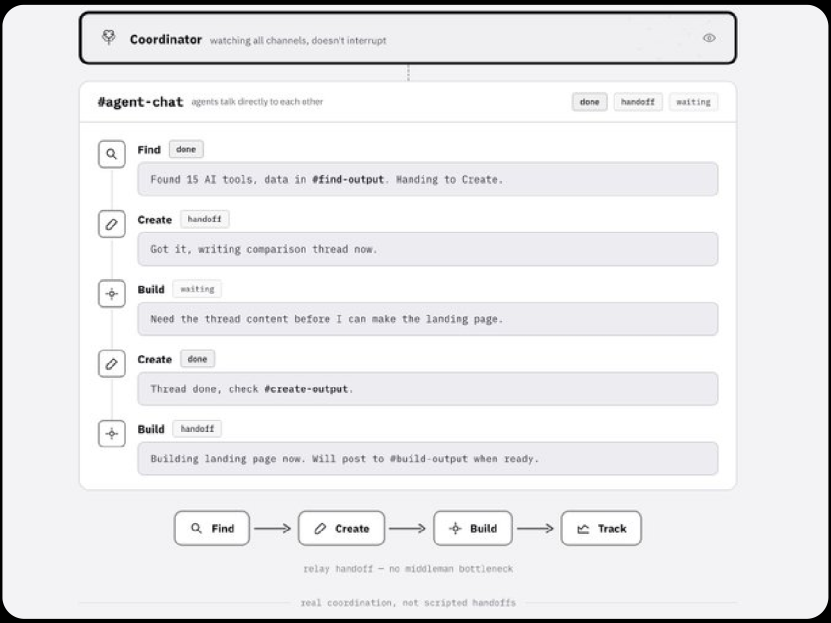
> **Find**: "Found 15 AI tools, data in #find-output, handing to Create"

> **Create**: "Got it, writing comparison thread now"

> **Build**: "Need the thread content before I can make the landing page"

> **Create**: "Thread done, check #create-output"

its a real coordination, not scripted handoffs..


Coordinator watching all channels, doesn't interrupt
#agent-chat agents talk directly to each other
done
handoff
waiting
Find
Create
Build
Track
relay handoff — no middleman bottleneck
real coordination, not scripted handoffs

Find done
Found 15 AI tools, data in #find-output. Handing to Create.

Create handoff
Got it, writing comparison thread now.

Build waiting
Need the thread content before I can make the landing page.

Create done
Thread done, check #create-output.

Build handoff
Building landing page now. Will post to #build-output when ready.

=========

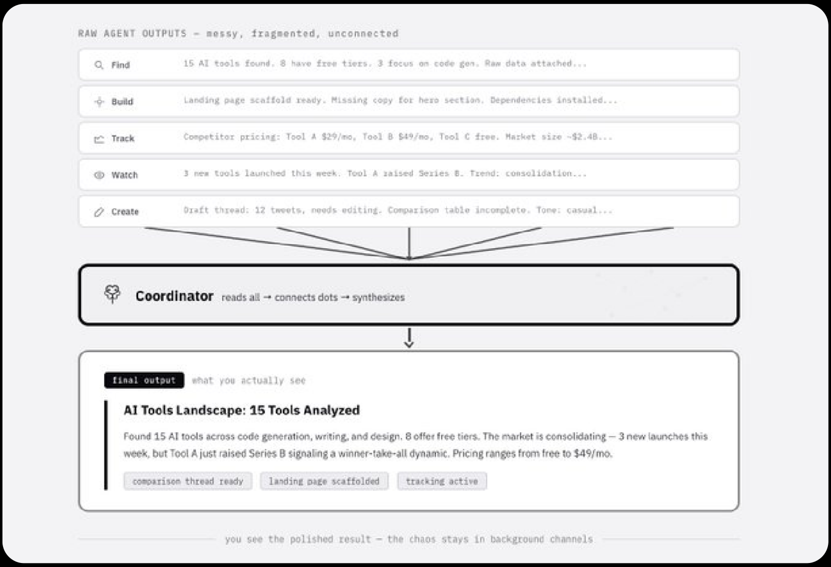
The Synthesis

Coordinator reads all outputs, combines into one clean result.

you asked one question, 5 agents worked on it, you get back one answer ,not 5 separate messages.

the magic happens in the synthesis and raw agent outputs are messy.. the coordinator cleans it up, connects the dots, presents it like a human analyst would.

you see the polished result so the chaos stays in the background channels.


RAW AGENT OUTPUTS — messy, fragmented, unconnected
Coordinator reads all → connects dots → synthesizes
final output what you actually see
you see the polished result — the chaos stays in background channels

Find 15 AI tools found. 8 have free tiers. 3 focus on code gen. Raw data attached...

Build Landing page scaffold ready. Missing copy for hero section. Dependencies installed...

Track Competitor pricing: Tool A $29/mo, Tool B $49/mo, Tool C free. Market size ~$2.4B...

Watch 3 new tools launched this week. Tool A raised Series B. Trend: consolidation...

Create Draft thread: 12 tweets, needs editing. Comparison table incomplete. Tone: casual...

AI Tools Landscape: 15 Tools Analyzed

Found 15 AI tools across code generation, writing, and design. 8 offer free tiers. The market is consolidating — 3 new launches this week, but Tool A just raised Series B signaling a winner-take-all dynamic. Pricing ranges from free to $49/mo.

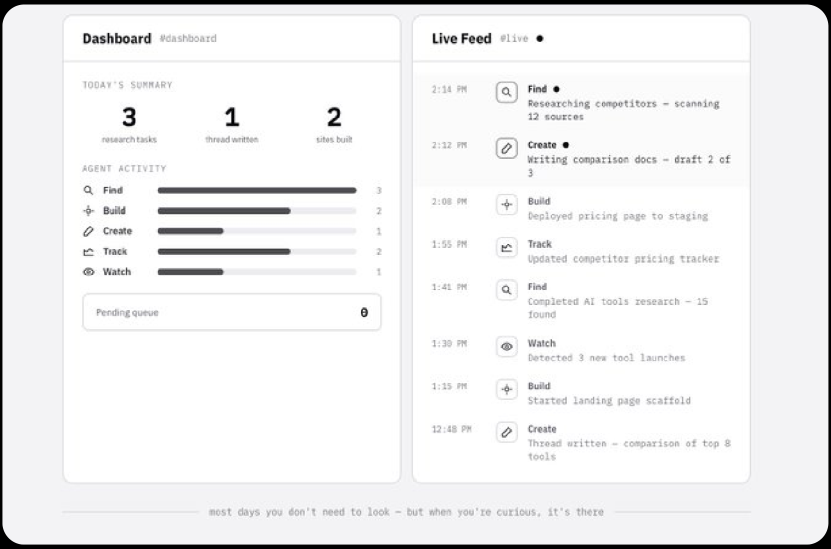
comparison thread ready | landing page scaffolded | tracking active

=========

Dashboard + Live Feed *(you need channels for this)*

Dashboard shows what happened today at a glance.. Live feed lets you watch agents work in real time.

most days you don't need to look. but when you're curious, it's there.
high level overview or deep dive.. your choice.



```
Dashboard #dashboard

TODAY'S SUMMARY
3                 1                2
research tasks    thread written   sites built

AGENT ACTIVITY
Find      3
Build     2
Create    1
Track     2
Watch     1

Pending queue    0
```

```
Live Feed #live ●

2:14 PM    Find ●
           Researching competitors – scanning 12 sources
2:12 PM    Create ●
           Writing comparison docs – draft 2 of 3
2:08 PM    Build
           Deployed pricing page to staging
1:55 PM    Track
           Updated competitor pricing tracker
1:41 PM    Find
           Completed AI tools research – 15 found
1:30 PM    Watch
           Detected 3 new tool launches
1:15 PM    Build
           Started landing page scaffold
12:48 PM   Create
           Thread written – comparison of top 8 tools
```

most days you don't need to look – but when you're curious, it's there

=========

# Memory System

Agents read their memory channels when they spawn. they know what they knew last time. They get smarter every run.

Two layers:

> Discord channels (shared team memory)
> Local .md files (private agent memory)

> **Find** agent remembers which websites have good data.
> **Create** agent remembers your writing style.
> **Build** agent remembers your preferred tech stack. Knowledge compounds automatically.

=========

# Local Config (Agent DNA)

SOUL.md defines personality.
AGENTS.md has the rules.
MEMORY.md stores knowledge.

This lives on YOUR machine, not in some cloud. The agent DNA is yours.

=========

# Always On

Heartbeats pulse every 30 minutes. Cron jobs handle scheduled tasks, event triggers react to the world.

You wake up to a morning brief of what happened overnight.

"Found 5 new competitors while you slept"
"Your thread got 50 replies, top feedback themes analyzed"
"Server metrics look good, no issues detected"

work continues when you're not there.

=========

# Researches (Self improvement)

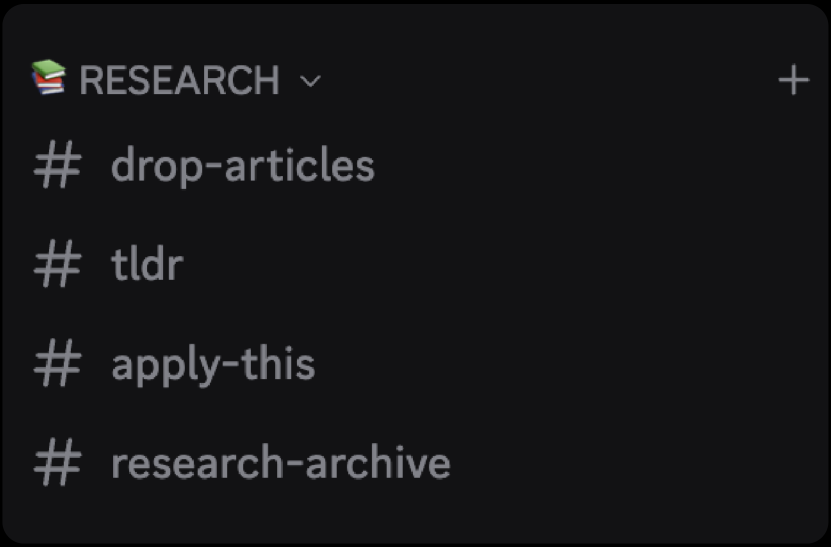
Drop a link in #drop-links. System auto summarizes, extracts takeaways, archives it.

Research on autopilot.

Every article you save gets processed:

- Summary in your words
- Key takeaways extracted
- Whatever we can apply in our system to improve
- Filed in searchable archive

  basically, the more articles and insights you find on X or other platforms and the more you drop in, the more your agents get smarter, because you're feeding them real knowledge they remember next time.. your input becomes their intelligence... the system learns what you learn.



📚 RESEARCH

drop-articles

tldr

apply-this

research-archive

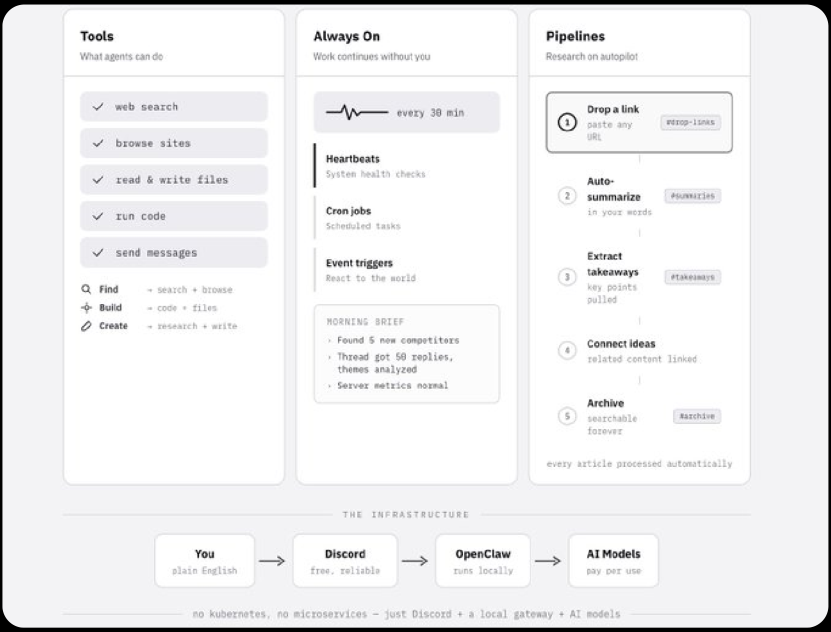
=========

# The Infrastructure

Discord handles coordination and persistence. OpenClaw gateway on your machine connects to AI model providers.

You ↔️ Discord ↔️ OpenClaw Gateway ↔️ AI Models

> Discord is free
> Reliable, and already built
> OpenClaw is open source and runs locally
> AI models you pay for usage.



Why Discord?

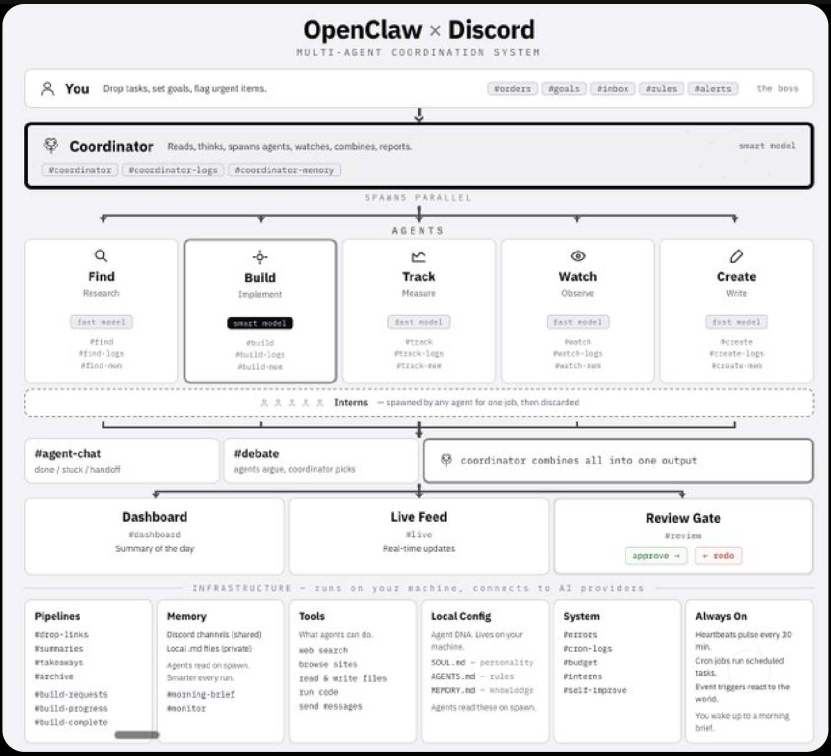
- Channels = workspaces
- Threading = deep work
- Permissions = access control
- Search = free
- History = unlimited
- Mobile = free app , easy monitoring

Everything you would spend weeks building already exists. Discord solved group communication at scale so you just repurpose it for AI coordination.

Plus you can set your agents to DM you directly when something urgent happens, you obviously can't do that with a database..

=========

# Full System Architecture



==============

How to start?

Step 1: Create Your Discord Server
Step 2: Get your Discord Bot Token.
Step 3: Connect Discord to OpenClaw
Step 4: Make sure your LLM, OpenClaw and Discord are connected
Step 5: Name Your Coordinator
Step 6: Give It Your First Task
Step 7: Add More Agents as You Need Them
Step 8: Let Your Coordinator Handle the Rest

# Final Thoughts

This is just orchestration, one coordinator and a few agents, very basic handoffs.

but this is where it's all heading, swarms, hundreds of agents coordinating in real time and splitting complex problems into micro-tasks to them in parallel

what I built is a simple version of what every company will run in two years.

agent coordination isn't a feature, it's the whole architecture, the models keep getting cheaper, the context windows keep getting bigger and the orchestration layer is what makes it all useful.

right now you can run at least 5 agents on Discord for almost nothing. the models get cheaper as we go.. be ahead because most people are still copy

what you want..thanks for reading.

JUMPERZ
@jumperz

Follow

UX Designer . Founder of @Rebirthstud_io , still trenching through every cycle smh