S2He Lab
AI Agent Share Session

Zhiyuan Wang | March 27, 2026 | Link Lab 211

A Day in My Life (AI-Powered)

Auto Parking

```
[parkmobile] Parking purchased!
Zone: 5556 (UVA-Red)
Vehicle: ANIUY (VA)
Duration: 9h 30m
Price: $1.3 (total w/ fees: $1.75)
```

6:53 AM — fully automated

Review AI News & Emails

```
Boo: 3 new arxiv papers on
  mobile health sensing
Boo: 2 emails need attention:
  → Reply: advisor re: defense slides
  → Be aware: lab meeting moved to
3pm
```

Morning briefing on phone

Research Work

AI agents review my papers and run experiments overnight.

```
/review-panel paper/methods.tex
# 3 simulated reviewers critique
# your paper before real ones do
```

```
Boo: [openclaw] Experiment update
# ToM A/B test batch 3 done
# Accuracy: 78.2% (+4.1%)
# Results pushed to repo
```

Show output

My AI Company

6 AI agents as employees. They code, review, research, and operate — 7/24.

CTO | CPO | COO

Product Shipping

AgentOS v0.3 pushed to GitHub
3 PRs merged overnight

Auto Research

Cognitive Bias Atlas of LLMs
Literature review: 42 papers scraped

More on this later...

All of this runs on **Claude Code** — let me show you how.

Today's Session

1. **Opening Hook** (you just saw it)
   Boo, dissertation prep, personal company — real AI-powered workflows

2. **Claude Code Deep Dive** ~20 min
   Sessions, skills & commands, MCPs, CLAUDE.md — the building blocks

3. **RyanHub iOS App** ~10 min
   Personal AI with 10 passive sensors — built with Claude Code, directly relevant to our research

4. **AgentOS: A Beta One Person Company** ~10 min
   6 AI agents as team members — "You are the leader!"

💬 Q&A and discussion throughout — interrupt me anytime

What is Claude Code?

A **terminal-native AI agent** that reads your codebase, plans changes, edits files, runs commands, and iterates — autonomously.

```
# Install
npm install -g @anthropic-ai/claude-code

# Start in any project
cd ~/your-research-project
claude

# That's it. You're in. 🚀
```

$ $5-15/mo light use | $200/mo Max plan (unlimited)

1M Tokens

Reads entire codebases. Understands cross-file deps.

Read/Write/Run

Edits files, runs tests, commits. Not just suggestions.

Agentic Loop

Plan → build → test → fix → repeat.

Extensible

Plug in any tool via MCP (more on this soon).

Sessions & Context

A **session** is a conversation with Claude Code. It has memory, context, and can run for hours.

1M Token Context Window

Your entire codebase in memory. Understands cross-file dependencies, architecture patterns, and data flow.

Session Continuity

Resume where you left off. Context auto-compresses for long sessions. `claude --resume`

Multi-Session Parallel

Run multiple agents simultaneously — one writes code, one reviews, one runs tests.

Headless Mode

Non-interactive: `claude -p "run all tests and fix failures"`

> *"I run ~**300 sessions per day** across my projects. Each session knows the project context instantly because of CLAUDE.md."*

~5,600

Sessions per week (my usage)

For researchers: Think of a session as a RA who already read all your code and papers. You just tell them what to do next.

Skills & Commands

Reusable prompts you define once, invoke with a slash — your personal AI workflow library.

Custom Commands

```
# Define once in .claude/commands/

/analyze data/sensor.parquet
# Your custom data analysis prompt

/review paper/methods.tex
# Your paper review workflow

/experiment configs/grid.yaml
# Launch experiment pipeline
```

Store in `.claude/commands/` — just Markdown files with prompts. Share across team.

Built-in Skills (my favorites)

/review-panel paper.tex

Simulates 3 expert reviewers critiquing your paper — catches issues before real reviewers do

/handoff

Creates a structured summary of current work for handing off to another agent or future session

/deep-research "topic"

Internet-scale investigation with references — returns an HTML report

Demo: /review-panel on my dissertation chapter — watch 3 "reviewers" critique it live

MCP: Model Context Protocol

A standard for plugging **any external tool** into your AI agent — databases, APIs, lab equipment, anything.

Apple Mail MCP

Claude reads and sends emails. "Draft a reply to Dr. Barnes about the meeting" — it reads the thread first.

▶ Demo

Overleaf MCP

Edit LaTeX papers on Overleaf directly from Claude Code. "Update Table 2 with new results." Compiles live.

▶ Demo

Telegram MCP

Agents notify me via Telegram when tasks finish, experiments complete, or errors occur.

GitHub

Create issues, review PRs, manage repos. Full GitHub workflow from your terminal.

Research Tools

arXiv, Semantic Scholar, PubMed. Literature review from your terminal.

Build Your Own

~100 lines of Python or TypeScript = custom MCP server. Plug in anything.

MCP is open-source by Anthropic — growing ecosystem of 1000+ community-built servers

CLAUDE.md: The Secret Weapon

A project memory file that makes every session **instantly effective**. Like onboarding a research assistant who **never forgets**.

```
# My Research Project — CLAUDE.md

## Setup
- Python 3.11, conda env: `paa-env`
- Run: python run_experiment.py
- Test: pytest tests/ -v

## Data Schema
- activity.csv: user_id, steps, calories
- readings.parquet: ts, value, type

## Constraints
- Use PyTorch, not TensorFlow
- All plots as PDF to figs/
- Black formatter, 88 char lines

## Current Work
- Working on V6 model (agentic pipeline)
- Deadline: CHI 2026 camera-ready
```

What to put in it

- **Build & run commands** — how to install, test, deploy
- **Project structure** — where things live, key files
- **Data schema** — column names, types, formats
- **Constraints** — "use PyTorch not TF", "plots in PDF"
- **Code style** — formatter, naming conventions

Pro tip: I have **16 projects** with CLAUDE.md files. Each session starts with full context — zero warmup.

Research Workflows in Action

Data Analysis

*"Load **sensor_readings.parquet**, compute per-participant stats, create a correlation heatmap."*

- Reads your data schema
- Writes pandas + matplotlib
- Runs & checks output
- Iterates on errors

Experiments

*"Grid search: **lr=[1e-3, 1e-4]**, **model=[lstm, gru]**, **seeds=[42, 123]**. Log to CSV."*

- Generates run scripts
- Error handling + logging
- Results CSV parsing
- Summary figures

Key insight: Claude Code doesn't just write code — it **runs it**, sees errors, and **fixes them autonomously**. You get the final working result.

RyanHub iOS

A personal AI super-app — built in 6 days with Claude Code

415K+ LOC | Zero external iOS dependencies | 100% self-hosted

"Change your mind: Apps are no longer there,
Personal AI app is coming.*"*

How to develop it with Claude Code? **Just chat with it and iterate!**

The Journey to RyanHub

Each problem I solved led to the next question — until a unified app became inevitable.

1

"Can I read AI-generated books?"

BookFactory — auto-generates & delivers books to my phone daily

2 books/day, 100+ generated

→

2

"Can agent buy parking before 7am?"

```
[parkmobile] Parking purchased!
Zone: 5556 (UVA-Red)
Duration: 9h 30m
Price: $1.75
```

6:54 AM — fully automated

→

3

"Can I monitor my health passively?"

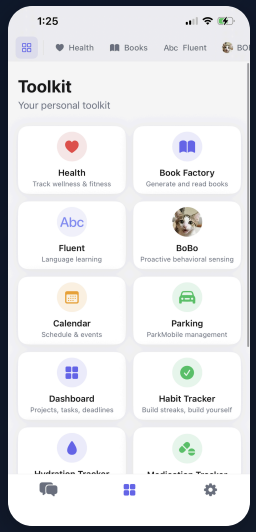
Apple Watch → auto-sync heart rate, steps, sleep to my AI

10 sensors, real-time streaming

→

4

"Why not one app for everything?"



415K+ LOC, built in 6 days

RyanHub: Hub-and-Spoke Architecture

A self-hosted iOS platform: AI chat at the center, **6 modules** that each enrich and read from a shared context bus.

AI Chat (Hub)

Ask anything — your AI knows your schedule, health, location, and activity. Answers are personalized.

Health & Wellness

Food logging (AI photo analysis), weight tracking, HealthKit integration. Trends + calorie analysis.

Bobo: Passive Sensing

10 passive sensors running continuously. Zero user burden. 7.4x data compression.

Calendar

Natural language scheduling. Google Calendar sync. AI knows your availability.

BookFactory

AI-generated books + audiobooks on any topic. Personal reading & learning platform.

Parking

Auto-parks at UVA. No more ParkMobile. Just works every morning.

▶ **Live demo:** Let me show you the app on my phone

The PersonalContext Bus

Every module feeds data into a **shared context bus** — the AI knows everything about your day without you telling it.

- Motion
- Health
- Location
- Screen
- WiFi
- Bluetooth
- Calls
- Audio

Bobo: 10 passive sensors → **7.4x compression** → behavioral timeline

> **"Am I getting enough sleep?"**
>
> Bobo: 6h 20m avg (below recommended)
>
> Health: 4 gym sessions, 7.2K steps/day
>
> Calendar: 5-6 meetings/day this week
>
> AI synthesizes: "You're 45 min short. High meeting load is likely the culprit."

No explicit wiring. Every module feeds the bus, chat reads the bus. Cross-module intelligence emerges automatically.

S2He Connection

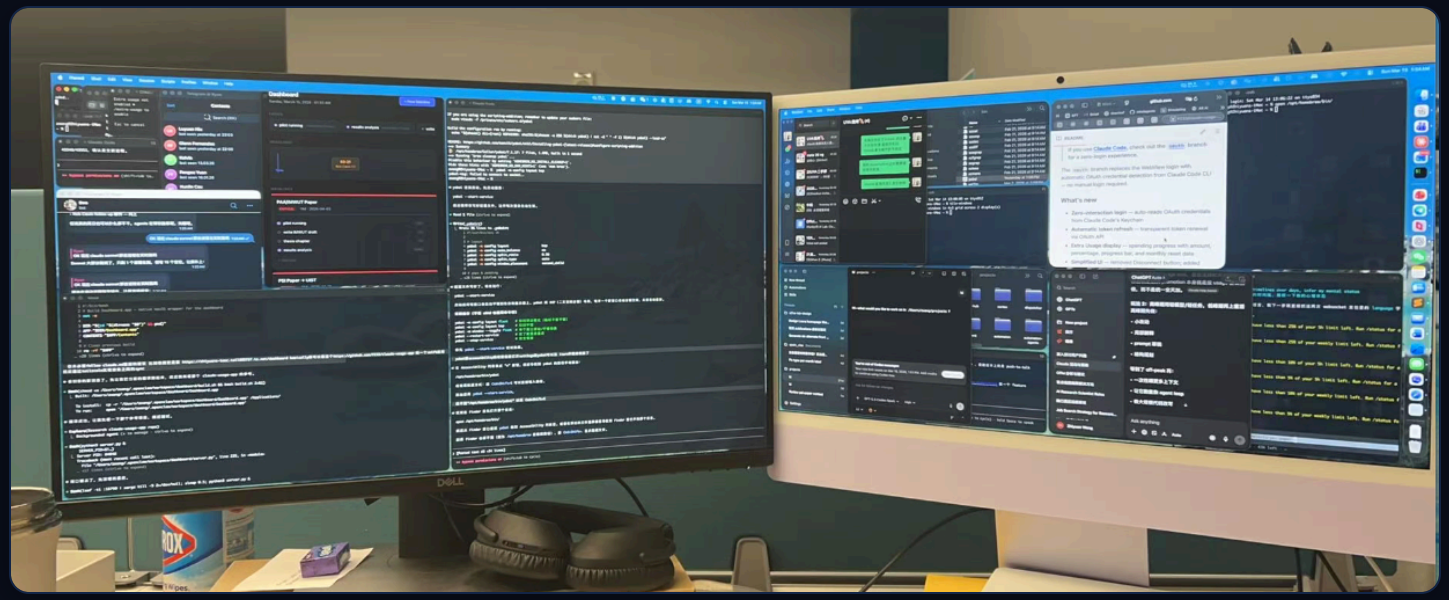
This is exactly what our lab studies — **sensing systems for health**. RyanHub is a living research platform, running daily on a real person's phone.

AgentOS

A Beta One Person Company

What happens when you push Claude Code to the limit?

Why Do I Need A Company Dashboard?



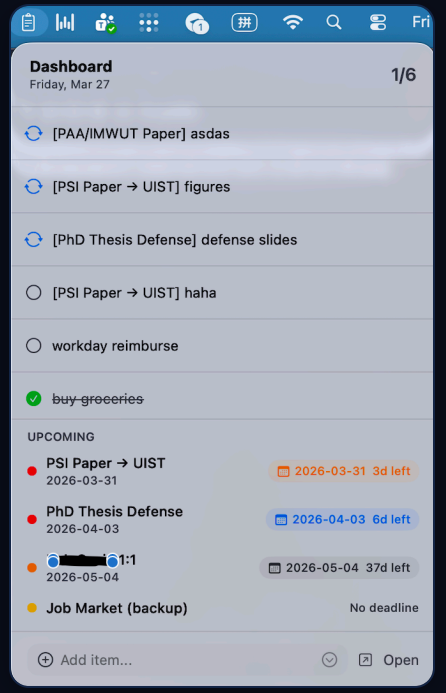
A "Modern Person's" Desktop

A New Working Paradigm...

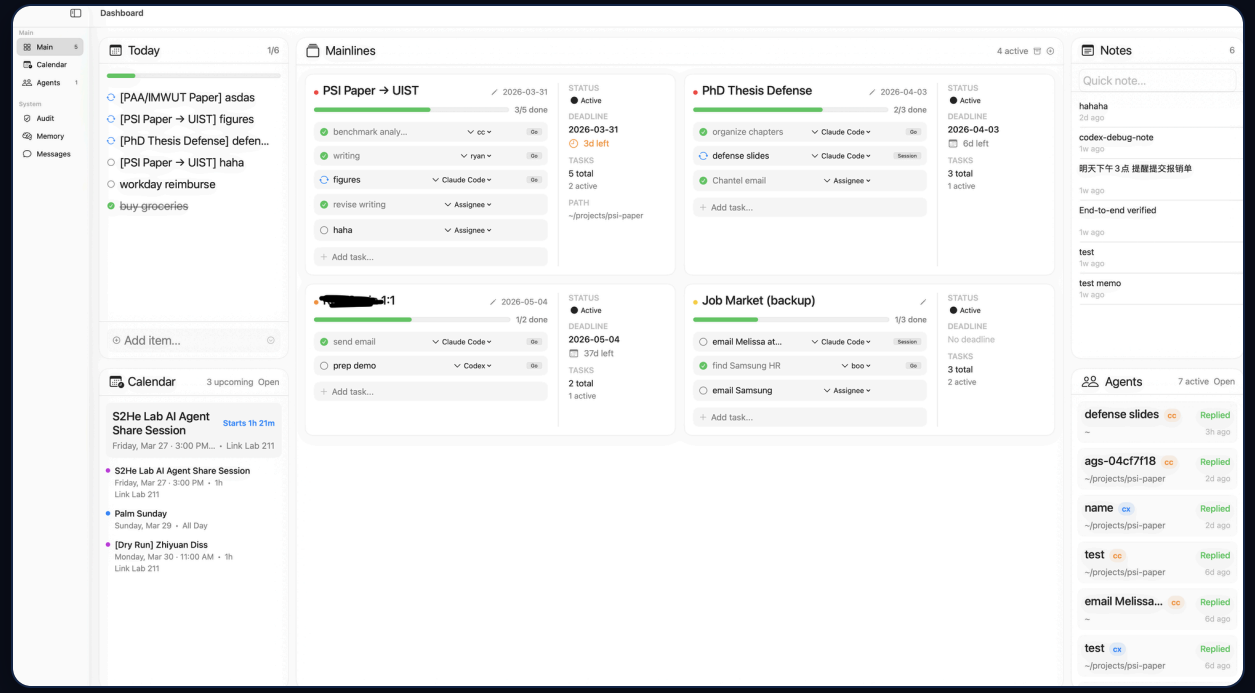
A Challenge to Human Cognition...

I Did My Best...

Streamline my workflow and reduce my cognitive load — **but not good enough.**



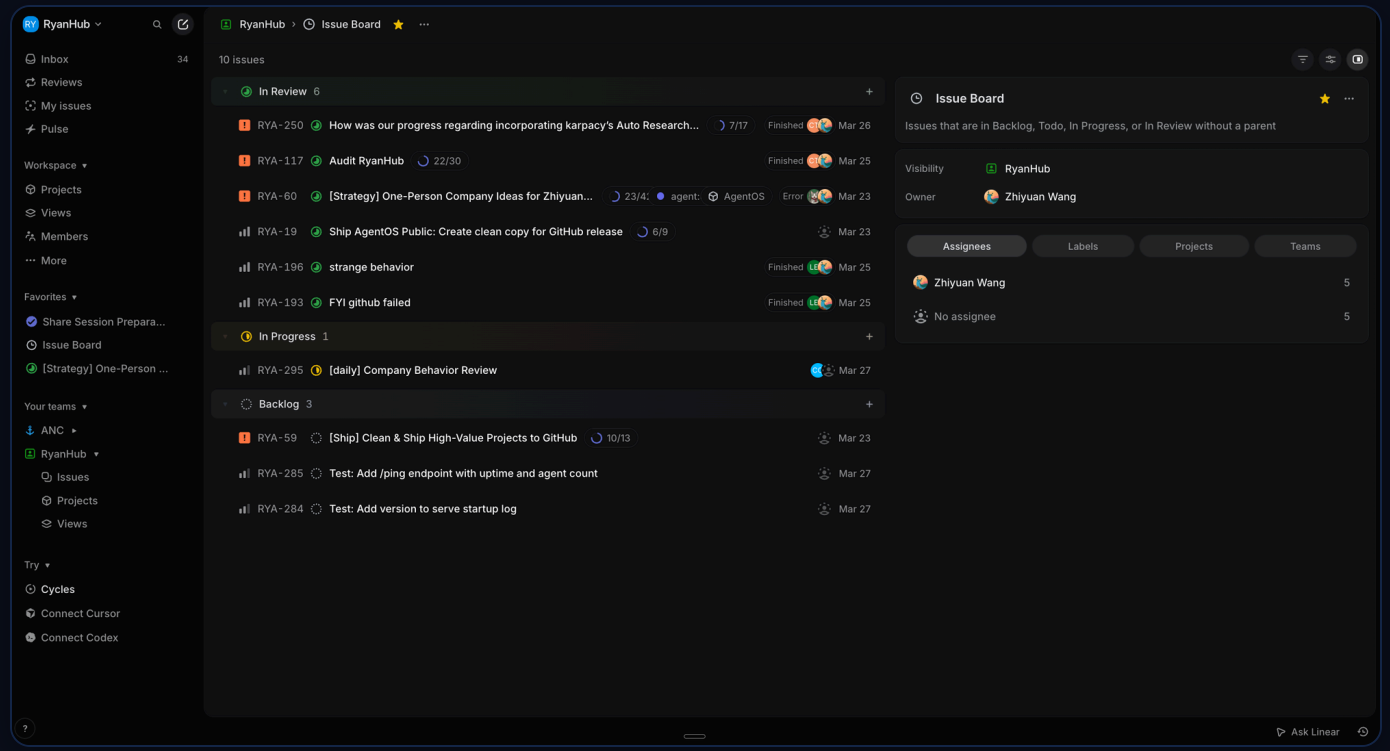
macOS Menubar Widget



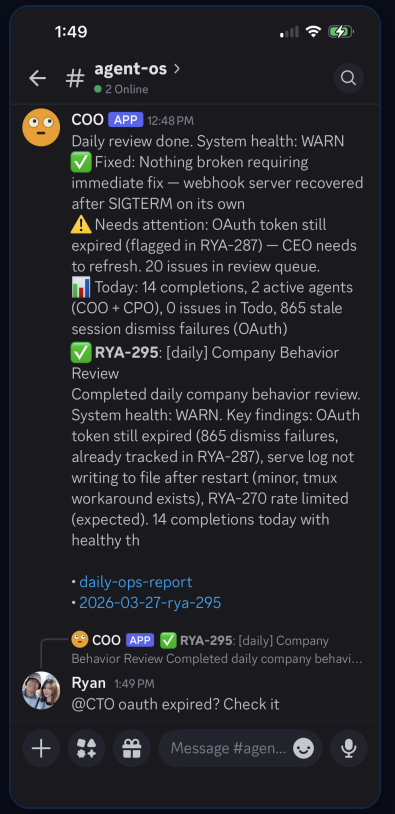
Unified Web Dashboard

I need a company!

Now I Am The CEO



Linear: Track issues & assign to agents

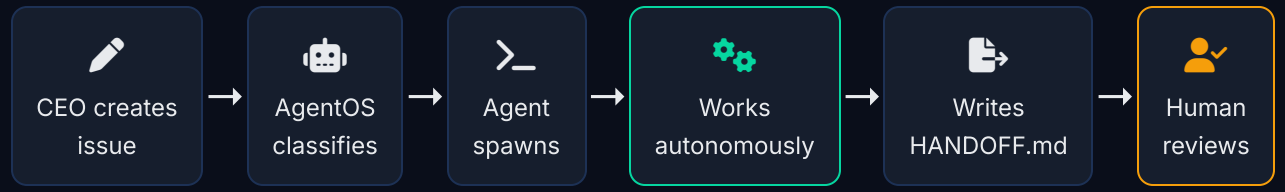


Slack: Agents report, CEO decides

The Agent Team

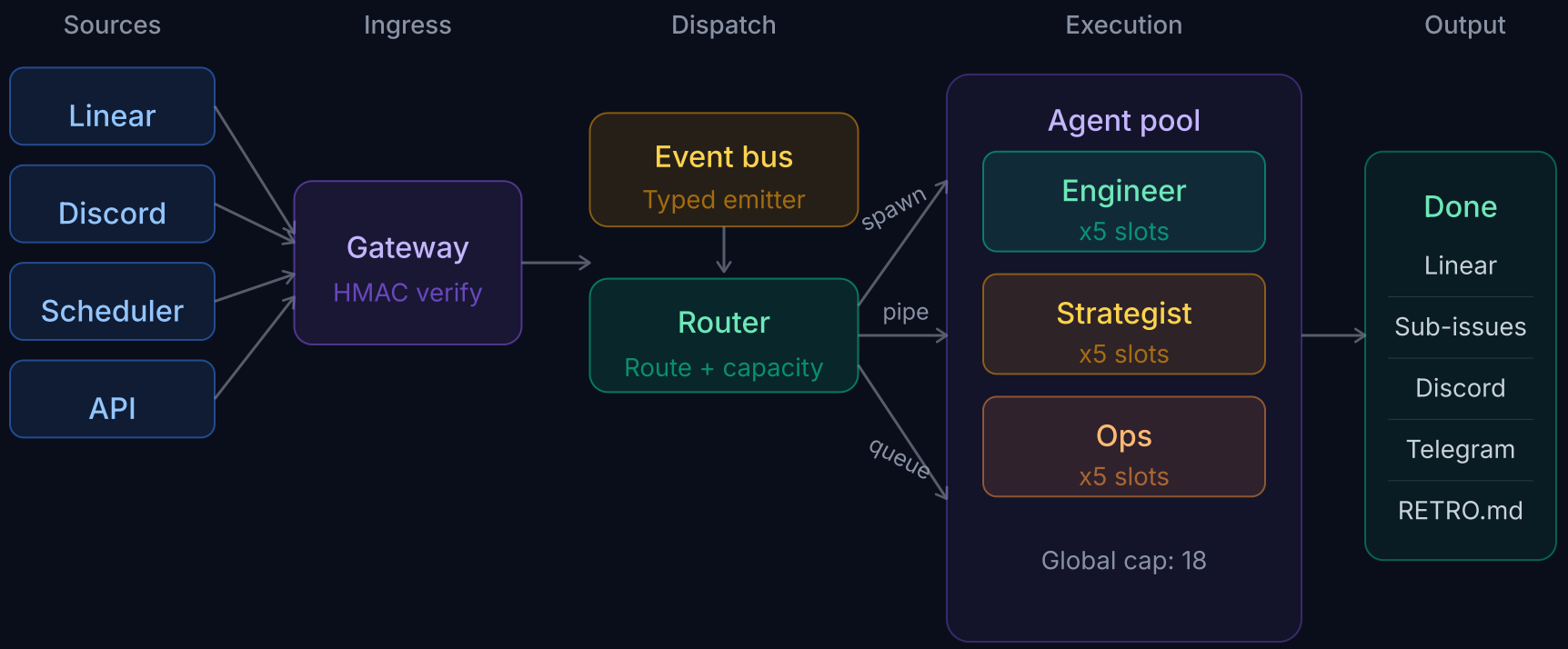
6 AI agents as real team members — persistent identity, institutional memory, tracked on Linear.

CEO Office	CTO	CPO	Lead Eng	Research	COO
Orchestrator & router	Architecture & QA	Product strategy	Feature builder	Literature & experiments	Operations



Death & Resurrection: Agents die after each task but resurrect with full memory via `.agent-memory/` files. No context window bloat — institutional knowledge accumulates across hundreds of sessions.

AgentOS Architecture



SQLite cache (disposable) — Linear is the single source of truth. Per-role memory persists via `~/.anc/agents/{role}/memory/`

By The Numbers

~300 Daily AI Sessions	**5,600** Weekly Sessions	**251+** Issues Completed	**31** Projects Managed
22K Lines of Code (AgentOS)	**564** Automated Tests	**1,580** Commits in March	**6** AI Agent Roles

Persistent Memory

Each agent writes `.agent-memory/` files that survive across sessions

3-Tier QA

Self-QA → CTO review → CEO approval. Every change passes three gates.

Auto-Research Swarm

32 parallel experiments with genetic selection. Finds optimal hyperparameters.

Real numbers from my system as of today. All agent-generated, human-reviewed.

You Are the Leader.

AI agents are your team — you provide the **vision**, they provide the **execution**.

Researchers

10x your data analysis, paper writing, experiment automation

Developers

Build full apps, debug complex systems, ship faster

Founders

Run an AI company with a team of 6 agents — as I showed you

Start with **one task** you've been putting off. That's how it all begins.

Q&A + Discussion

Mindset Shifts

Executor → Leader

Stop writing code 100%. Start directing agents. Your role shifts from doing to deciding.

Tools → Systems

Don't use one tool for one task. Build unified systems where shared context creates emergent intelligence.

Sessions → Institutions

AI agents aren't disposable chatbots. With persistent memory, they accumulate expertise across hundreds of tasks.

Ask Zhiyuan anything!