

Local Models & Self-Evolving Harnesses

Machine Learning Singapore #MSLG

- [Martin Andrews](#) = Harnesses @ [mdda.net](#)

25-June-2026

About Me

- Machine Intelligence / Startups / Finance
 - Moved from NYC to Singapore in Sep-2013
- 2014 = 'fun' :
 - Machine Learning, Deep Learning, NLP
 - Robots, drones
- Since 2015 = 'serious' :: NLP + deep learning
 - Including Papers...
 - & GDE ML; ML-Singapore co-organiser...
 - & Red Dragon AI...



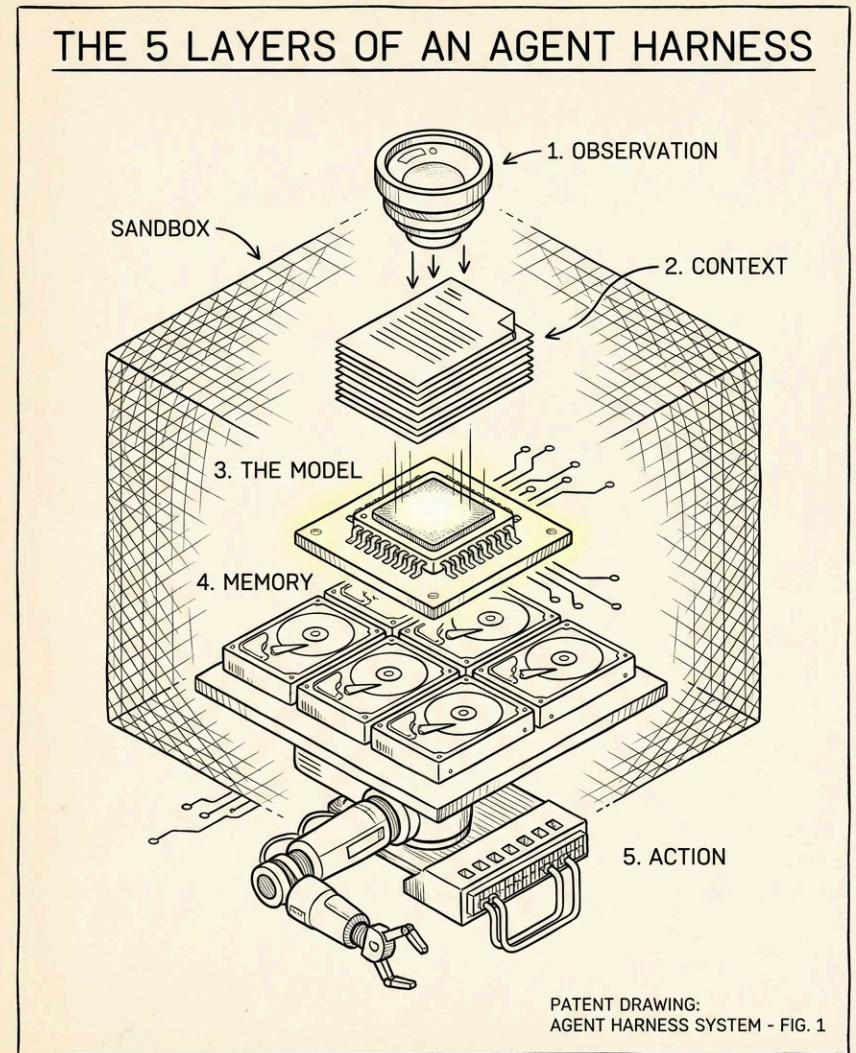
Outline

- Basic Harnesses
- Self-Modifying Harnesses
- Token Sourcing Risks
 - ... and mitigation
- Wrap-up & QR-code

Basic Harnesses

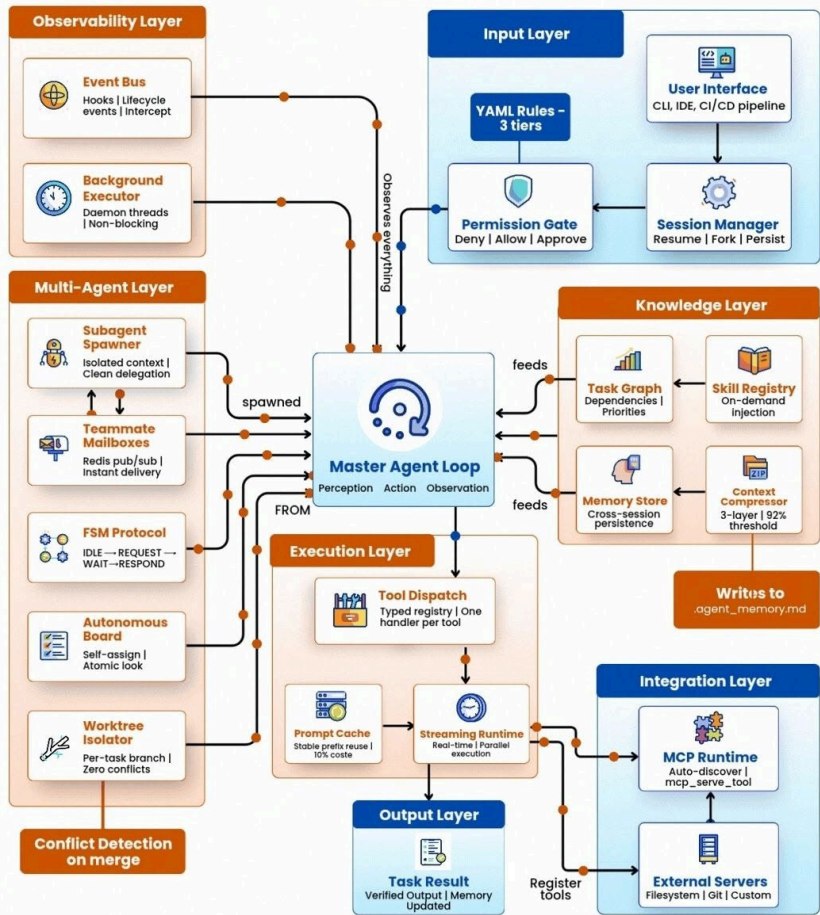
What is a Harness?

- Harnesses provide:
 - software infrastructure
 - runtime environment
 - engineering layer
- Wraps around a raw LLM to make it:
 - functional, reliable, and usable
- i.e. : Everything except the model itself



CLAUDE CODE ARCHITECTURE

Greg Coquillo
Product Leader



Claude Code

- Coding agent system
- Can launch subagents
 - write tests
 - loop until 'satisfied'
- Can also author new SKILL.md files
 - but this is an extra request

AutoResearch

- Started last year
 - Darwin Gödel Machine (May-2025)
 - Sakana AI
 - AlphaEvolve (June-2025)
 - DeepMind : A coding agent for scientific and algorithmic discovery
 - GPU Kernel Scientist (June-2025)
 - AMD GPU kernels...
- Karpathy meme version (March-2026)
 - simple repo deployment FTW
 - slick git branching for experiments
 - target : Karpathy's own `nanochat` experiments



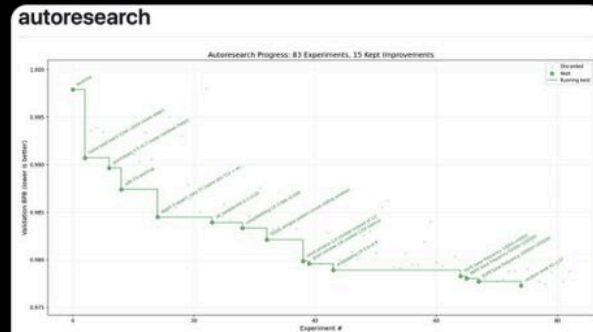
I packaged up the "autoresearch" project into a new self-contained minimal repo if people would like to play over the weekend. It's basically nanochat LLM training core stripped down to a single-GPU, one file version of ~630 lines of code, then:

- the human iterates on the prompt (.md)
- the AI agent iterates on the training code (.py)

The goal is to engineer your agents to make the fastest research progress indefinitely and without any of your own involvement. In the image, every dot is a complete LLM training run that lasts exactly 5 minutes. The agent works in an autonomous loop on a git feature branch and accumulates git commits to the training script as it finds better settings (of lower validation loss by the end) of the neural network architecture, the optimizer, all the hyperparameters, etc. You can imagine comparing the research progress of different prompts, different agents, etc.

github.com/karpathy/autor...

Part code, part sci-fi, and a pinch of psychosis :)



One day, frontier AI research used to be done by meat computers in between eating, sleeping, having other fun, and synchronizing once in a while using sound wave interconnect in the ritual of "group meeting". That era is long gone. Research is now entirely the domain of autonomous swarms of AI agents running across compute cluster megastructures in the skies. The agents claim that we are now in the 10,205th generation of the code base, in any case no one could tell if that's right or wrong as the "code" is now a self-modifying binary that has grown beyond human comprehension. This repo is the story of how it all began. -@karpathy, March 2026.

OpenClaw

- Personal Agent loop = Inspirational

- fka ClawdBot ~Jan-2026

- ClawCon = 4-Feb-2026 in SF

- MLSSG night = 26-Feb-2026

- Peter Steinberger now an OpenAI employee

- caused a run on the Mac-Mini

- security issues for the unwary

- ... says Cyber Security Agency of Singapore

- Still lots of hype around the project

- there's even an event on tonight!

- "No Technical Background Required"



Featured in Singapore

Hermes

- New entrant from Nous Research
 - [Project Page](#), and MIT licensed code repo
 - Pronunciation:
 - "noose" & "her-meess", not "noo" & "ermez"
- Key Features
 - a 'better' OpenClaw
 - tasteful software choices
 - batteries included (but disabled by default)
 - self-modification
 - automatically reviews what went well
 - and updates itself for future you

Top Coding Agents		View all →
1.	Hermes Agent Hermes Agent is an open-source, self-improving AI a...	1.03T tokens
2.	Kilo Code Kilo Code is an open-source AI coding agent that work...	246B tokens
3.	OpenClaw OpenClaw is an open-source AI agent that connects to...	184B tokens
4.	Claude Code Claude Code is Anthropic's agentic coding tool that rea...	142B tokens



HERMES - AGENT

THE COMPLETE GUIDE FOR EVERYONE

The self-improving AI agent that learns, remembers, and works for you 24/7.



Learns & Improves
Creates skills from experience and improves them.

Remembers Everything
Tree-tier memory that persists across sessions.

Works 24/7
Runs anywhere. Accessible everywhere via multiple platforms.

1 WHAT IS HERMES AGENT?
An autonomous AI agent with a built-in learning loop. It remembers, creates skills, improves them, and builds a deepening model of who you are across sessions.

Self-Evolving Skills Agent writes, refines, and reuses its own skills.	Runs Anywhere Local, Docker, SSH, Modal, Daytona, Singularity.
Tree-Tier Memory Facts, conversations, and decisions persist across sessions.	Multi-Model Support OpenRouter, OpenAI, Claude, Gemini, Llama, Local & more.
GEPA Optimization Goals auto evolution using execution traces.	Multiple Platforms CLI, Telegram, Discord, Slack, WhatsApp & 20+ more.

Hermes packages a gateway around a learning agent. OpenCrux packages an agent around a messaging gateway.

2 HOW IT'S BUILT
Everything flows through one core: AI:Agent class

```

System Prompt → Build → Check Goals → API Call (Interruptible) → Execute Tools → Loop (Up to 50 Turns)
    
```

5 Terminal Backends
Local • Docker • SSH • Modal • Daytona • Singularity

Universal Model Support
One command to switch between 200+ models and providers.

50-Turn Safety Cap
Prevents infinite loops and runaway costs.

3 BEFORE MEMORY: WHO IS THE AGENT?
SOUL.md defines identity, tone, and core principles. It's the first thing in the system prompt (50k #).
Hand-authored and static
Applies across all prompts and sessions
Sets the lens for memory and skills

Without identity, every agent feels the same. SOUL.md makes each agent uniquely yours.

4 THE MEMORY SYSTEM: THREE TIERS
Three layers, each built for a different purpose.

TIER 1	Core Memory (always in context) KeepAssistant (2,000 turns) • UllM84.md (2,399 chars) Persistent facts about projects you do	Fast Tiny Essential
TIER 2	Session Search (FTSS) Full-text search across all past conversation with LLM summarization.	Searchable Unlimited On-Demand
TIER 3	External Memory (Provider) 8 pluggable providers: Notion, Obsidian, Room, MongoDB, OpenAI, PineDB, Pg2, Chroma	Deep Persistent Indexed

Critical facts in Tier 1. Everything else searchable. Deep memory optional.

5 SELF-EVOLVING SKILLS
The agent writes its own playbooks and keeps them sharp.

```

skills.md
- Fix (learning description): Quick training plan (1-3 steps)
- author: agent

# GEPA
- Get goal status
- Check events
- Fail logs
- Review performance flag
- Review facts
- Add learning to skill

Skills are Markdown + YAML.
Progressive Disclosure
Level 0: Name + description only (~3 lines)
Level 1: Level 0 full skill when needed
Level 2: Drill into references

Self-Improvement Loop
Solve problems → Save as skill → Reuse next time → Get better

The Curator
Garbage collects old skills.
Auto-archives static skills. Never auto-deletes.
    
```

6 GEPA: EVOLVING SKILLS OFFLINE
Genetic-Parse Prompt Evolution using execution traces.

- Read Current Skill
- Generate Dataset
- Run GEPA Optimizer
- Evaluate Candidates
- Apply Constraints
- Fit with Best Variant

Why GEPA?
 Agents self-select poorly (self-congratulation bias)
 Prevents skill regression
 Offline optimization = no runtime cost
 PR-based = safe, reviewable, reversible

No GPU required.
Costs: \$1-10 per run.
Published at ICLR 2024.

7 GETTING STARTED
From solo AI sidekick to a team of agents that run everything.

```

Install pip install hermes-agent[all] or clone repo
Setup Wizard hermes init
Start Chatting hermes chat
Connect Telegram Set HERMES_BOT_TOKEN and use @.hermes_bot

What Lives in ~/.hermes/:
skills/ → your skill library
sessions/ → MBMM/FTSS/UBMM
memory/ → long-term knowledge
local.md → your identity & tone
config/ → models, APIs
providers/ → Notion, Obsidian
logs/ → traces & events
prompts/ → templates & system
    
```

8 GOING FROM 1 TO 10 AGENTS

```

Create / Name hermes create my-agent
Add / Manage Users hermes create user-agent
Define SOUL.md One agent per purpose
Schedule Work One or Interval execution
Let Them Work Anything runs 24/7 (agent never sleeps)

Example Team: Designer (Prod) • Programmer (Prod) • Researcher (Hobby)
    
```

9 CUSTOMIZING YOUR AGENTS

Programmer	• Python • React • Node • Code linting • Prettier • ESLint • Docker • State 10 (Database, Search, etc.) • Test (Unit, E2E, Smoke, Search, etc.)
Engineer	• System-level • Scaled thinking • Deployment strategy & reviews • Builds, Rollups, Releases & reviews • Code, APIs, error logs & metrics • Insight extraction • References • Multi-model systems • Clusters

10 CRON: SCHEDULING MADE SIMPLE
Describe what you want in English. Hermes handles the schedule.

```

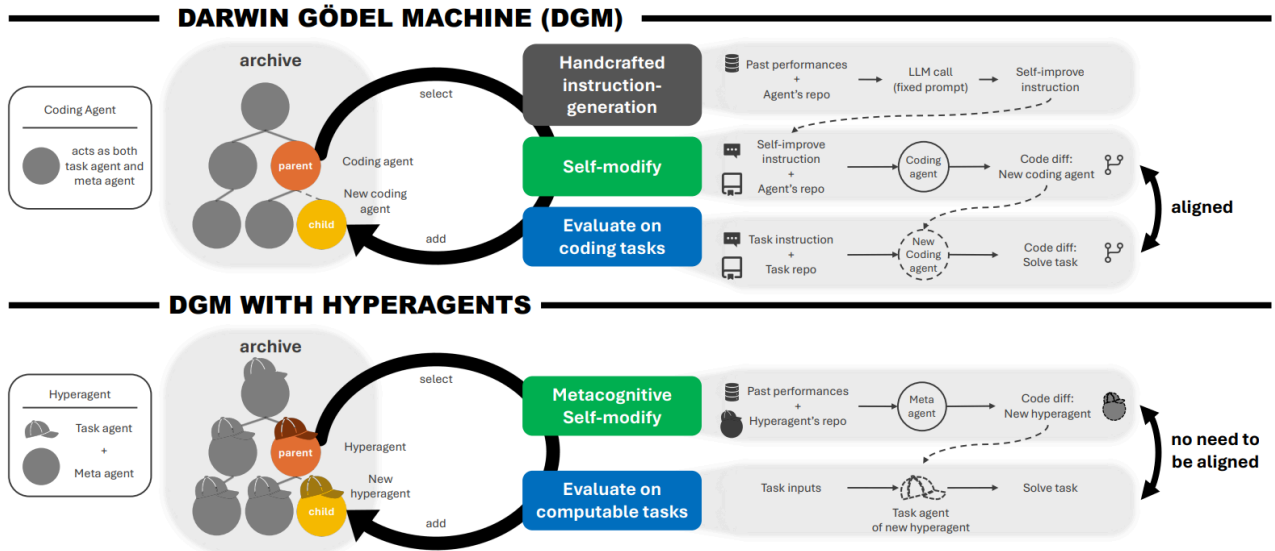
Describe Work "Generate test report"
Add Job "Every day at 10"
Set Constraints "Skip if not 10:00 - 11:00"
Define Target "Log to Notion" or "Email to Team" (self-reminder)
Check Logs "Show me runs, summary, top failures?"

Remember Examples
Every schedule is a skill.
Hermes 24/7 while you sleep
No GPU needed
No Cron setup
Built-in Telegram
Full Trace
    
```

Self-Modifying Harnesses

Darwin Gödel Machine - Hyperagents

- Hyperagents - Zhang *et al* (03-2026)
 - Jenny Zhang @ Meta (prev MLSG speaker)
 - with Jeff Clune
 - Facebook Research [Code Repo](#)
- Expands "AutoResearch" in a Meta direction
- Evolves an agent to do the task
 - and the update harness too
 - i.e. can specialise harness to task too
 - hence the 'Hyper level'
 - Can choose to add:
 - memory; performance tracking; etc ...



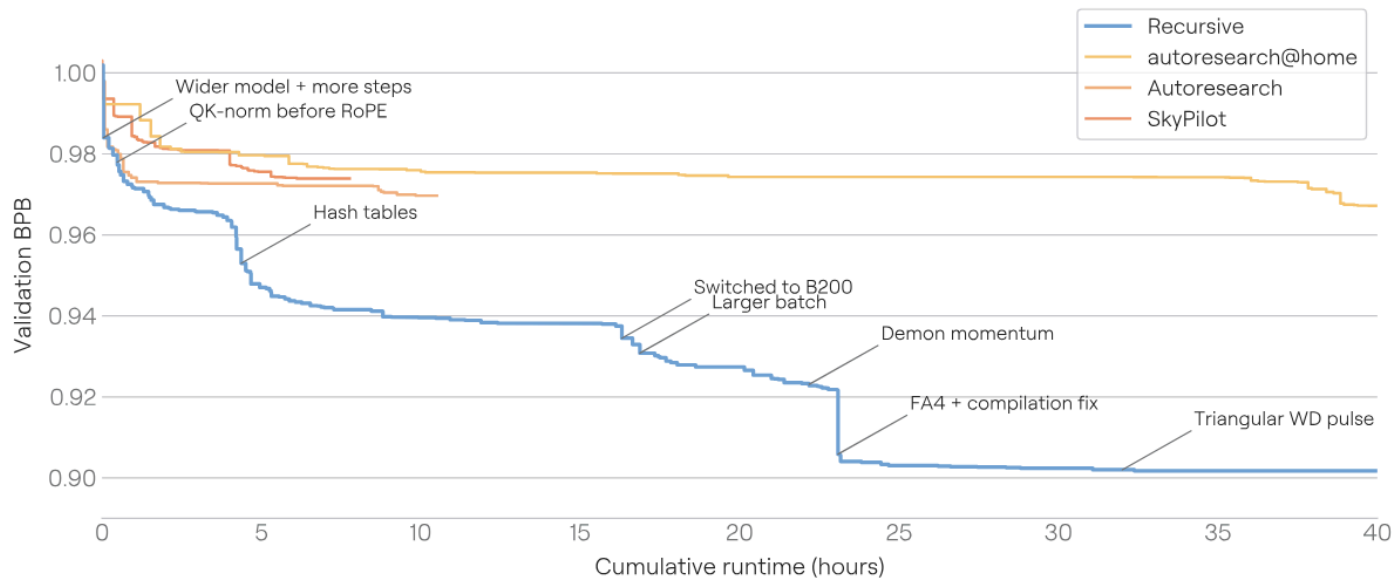
DGM-HyperAgents ++

- Jenny Zhang (and others) now at Recursive Superintelligence, Inc.
 - RSI raised \$650 million Series @ \$4.65 billion post-money

NanoChat Autoresearch: best validation BPB over wall-clock time

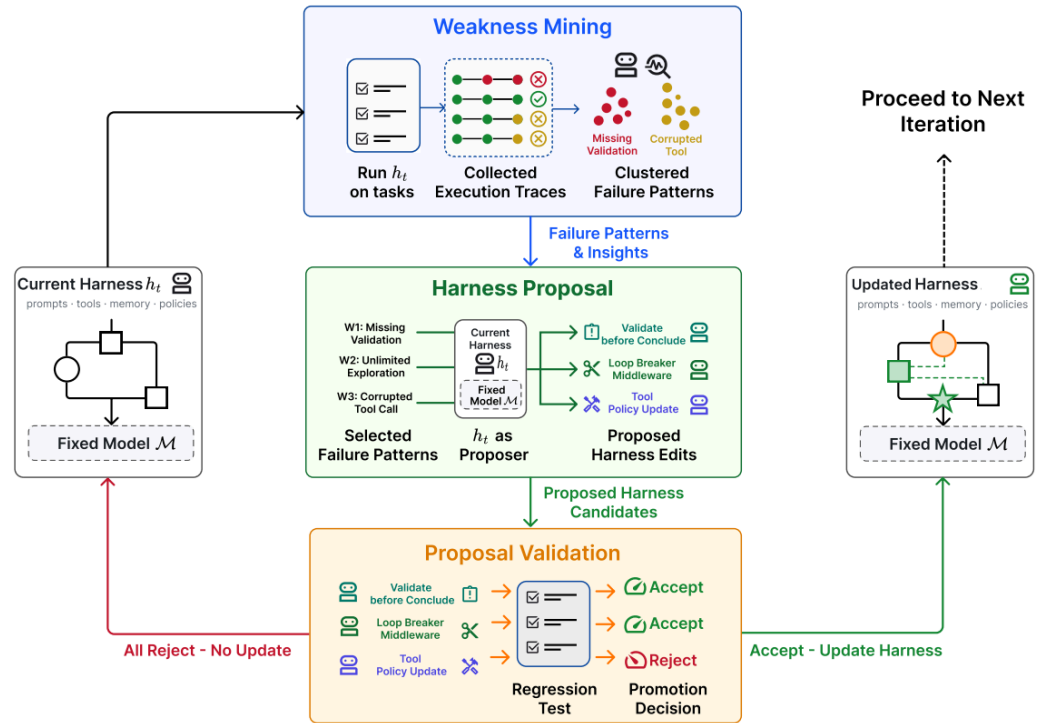


Recursive vs. Karpathy, SkyPilot, and autoresearch@home · single-seed evaluation · lower is better



Meta-Harness / Self-Harness

- Meta-Harness: End-to-End Optimization of Model Harnesses
- Lee *et al* (03-2026)
 - with Omar Khattab (DSPy + Recursive Language Models)
 - and Chelsea Finn (Stanford)
- Self-Harness: Harnesses That Improve Themselves
- Zhang *et al* (06-2026)
 - a framework that lets AI agents rewrite their own rules
 - boosting performance up to 60%

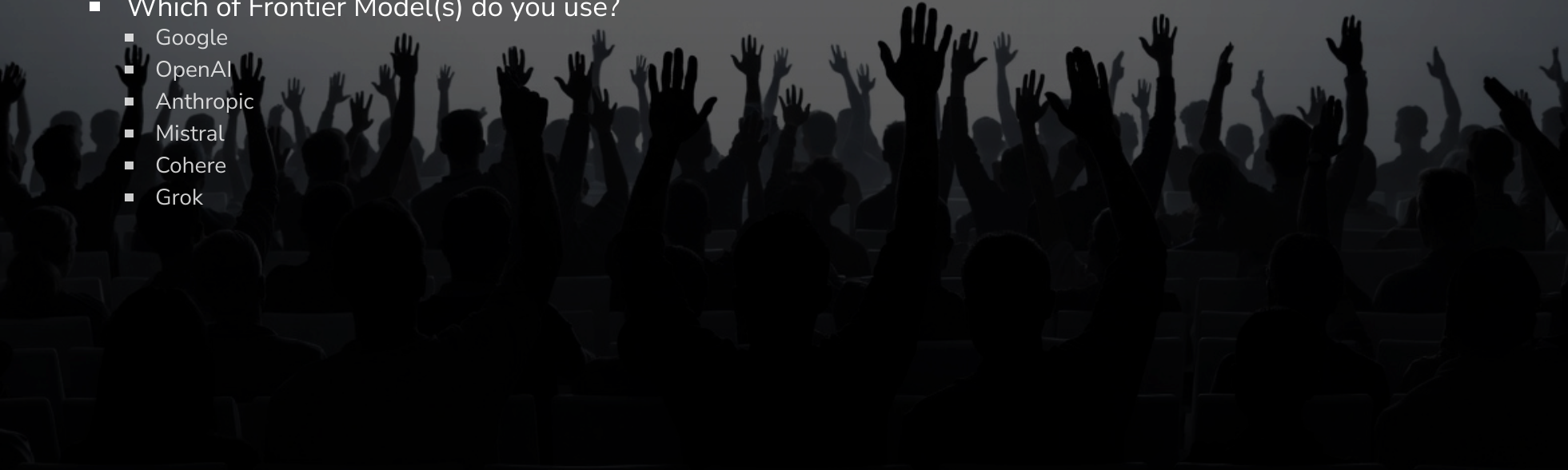


Token Sourcing Risks

Poll : Use of Frontier Models

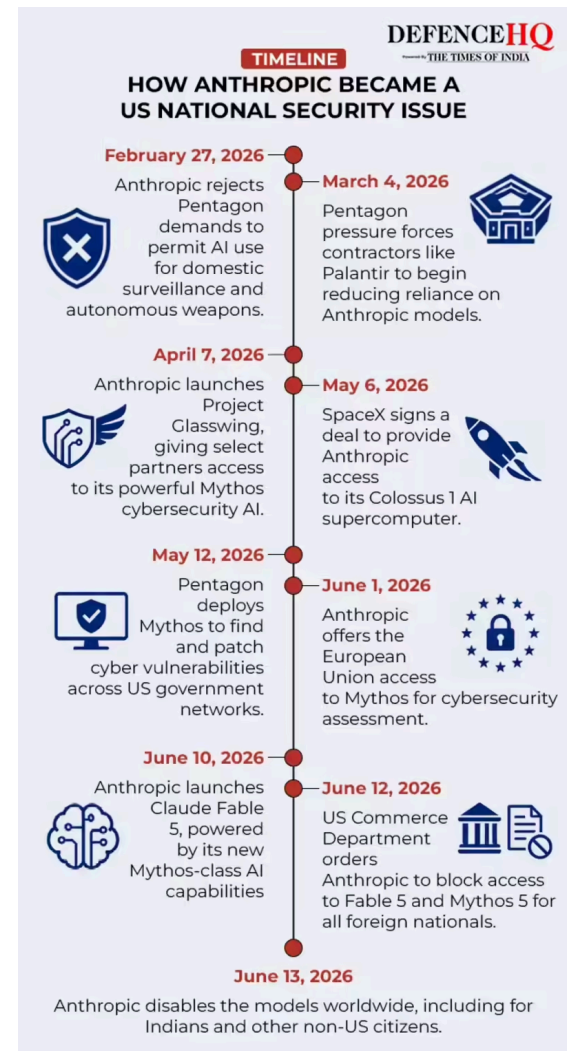
= Show of hands

- Which of Frontier Model(s) do you use?
 - Google
 - OpenAI
 - Anthropic
 - Mistral
 - Cohere
 - Grok



The Mythos/Fable saga

- Walk through Anthropic's timeline:
 - defense department
 - OpenAI takes up 'lawful use' position
 - Mythos model finds security issues
 - alarmingly many...
 - Fable released with guardrails
 - not just refusal!
 - AI researchers now feel the pinch
 - Amazon finds 'jailbreak'
 - enough to give leverage to US Government
 - Now blocked for everyone!
- PS: Don't assume OpenAI is safe choice ...



Owning Your AI

Who should be thinking about this?

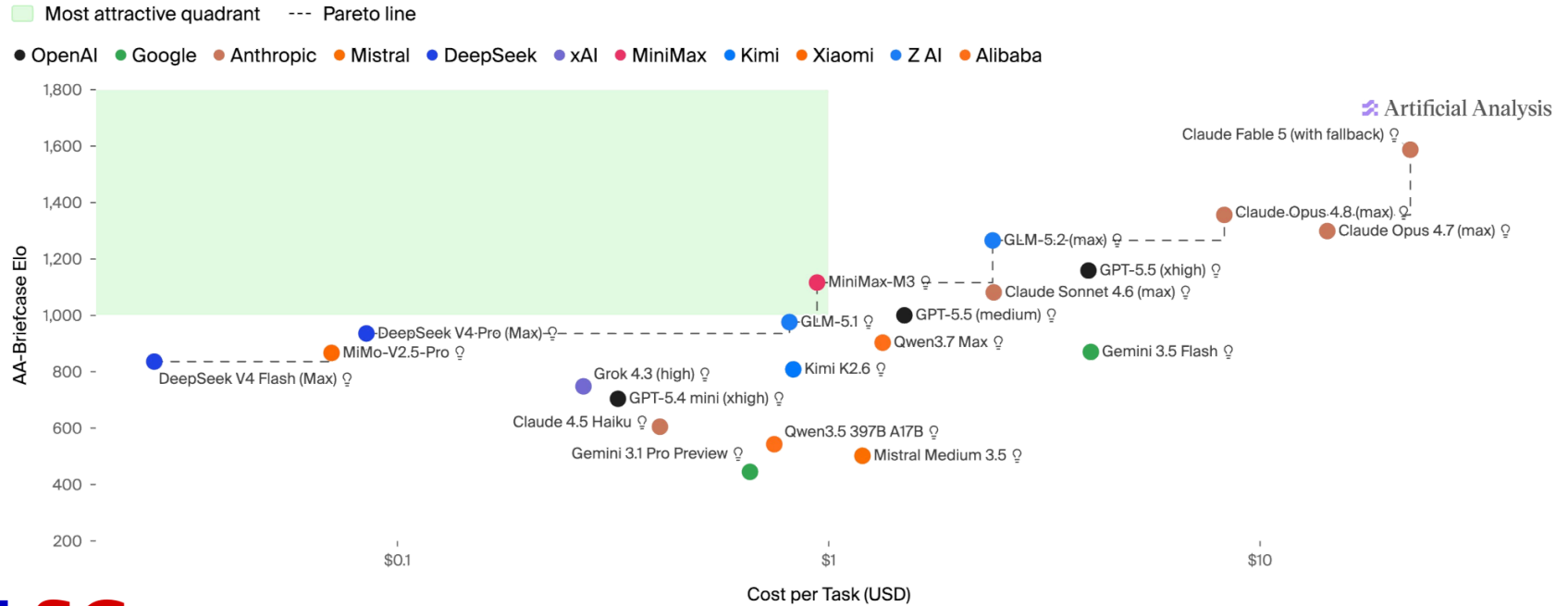
- Who?
 - Individuals
 - Companies
- Why?
 - cost
 - privacy
 - IP leakage
 - security
 - speed
- Where?
 - cloud (multiple providers?)
 - local (fully costed?)



Model capability / cost frontier

AA-Briefcase Elo vs. Cost per Task

AA-Briefcase Elo · Cost per task (USD)



Poll : Use of Open Models

= Show of hands

- Which families?

- Gemma
- Llama
- Qwen
- DeepSeek
- *other?*

- Sizing (parameter count)

- size <6B
- 6B < size <15B
- 15B < size < 40B
- 40B < size

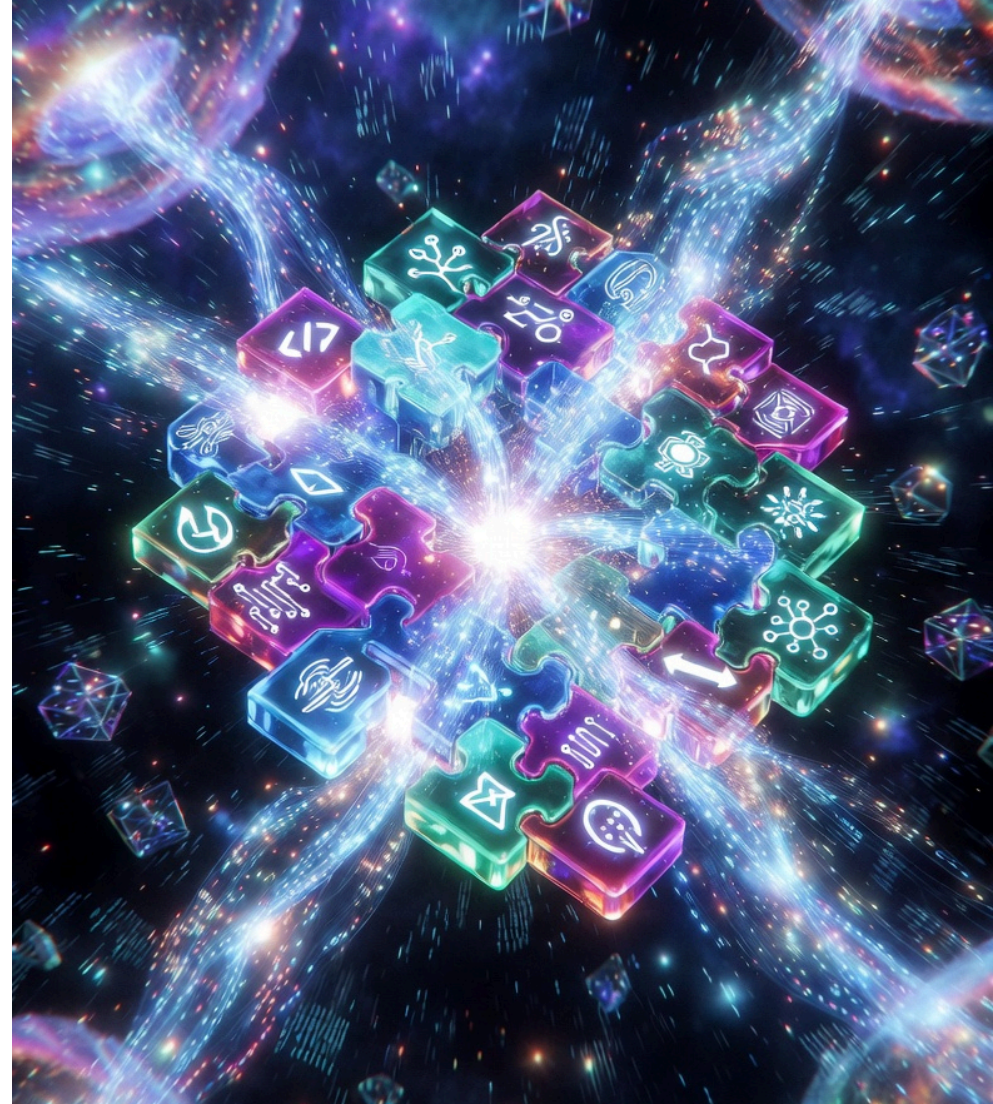
- Training

- None = off-the-shelf (+prompting)
- Supervised-Fine Tuning
- Reinforcement Learning



Integrating Open Source Models

- Problems with Foundation company lock-in:
 - cost
 - privacy
 - and now : country risk
- Solution
 - Use different models for different tasks
 - eg: Foundation for planning
plus Open Source models *increasingly*
 - Can do prompt / harness optimisation for both
 - But, with Open Source, you can:
 - train model on your local hardware
 - or in the cloud
 - train smaller models for repetitive tasks
 - for even bigger savings



Who agrees?

- Satya Nadella (Microsoft CEO)
 - published an article on Twitter/X
 - now advocating for an ecosystem approach
- Frontier Tuning: Teaching AI to work the way you do
 - announced by Microsoft on 2-June-2026
 - (no, not Open Source)
- Clearly, there's some tension with OpenAI...
 - and the Microsoft story makes sense
 - ... particularly with Open Source models

Satya Nadella reveals why every company may need its own AI model: the model becomes the new company database.

"To me, a model is like the database market."

"A firm should be able to take the tacit knowledge it has and embed it inside weights in a model that they control."

"When somebody asks me how many models should there be, I'll say as many models as firms in the world."

The contrarian part: the value may not sit in one universal frontier model. It sits in each company turning its private operating knowledge into a controlled model.



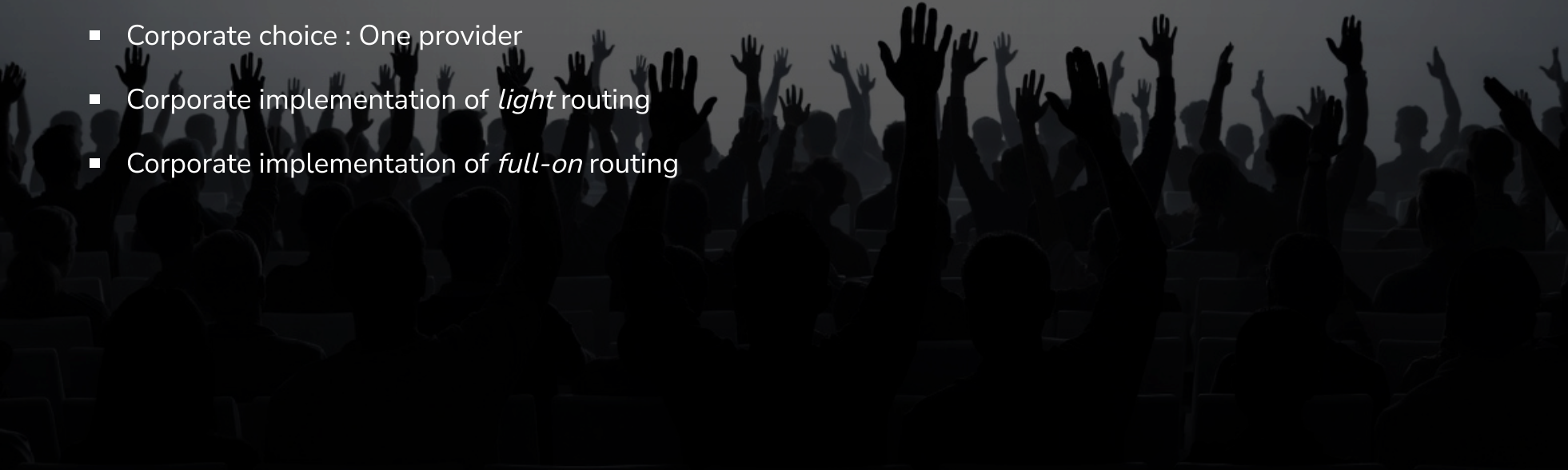
'Simple' Router approach

- Foundation model choices
 - which model
 - model capability per brand
- Monitor
 - tokens spent / by whom
- Routing
 - multiple sourcing
 - Personally Identifying Information redaction
- More capable *full-on* version
 - understanding caching
 - manage routing more carefully
 - usefulness of answers
 - = training data



Poll: Use of Router

- Just use personal accounts
- Corporate choice : One provider
- Corporate implementation of *light* routing
- Corporate implementation of *full-on* routing



EvoTrainer

- EvoTrainer: Co-Evolving LLM Policies and Training Harnesses for Autonomous Agentic Reinforcement Learning - Chen *et al* (2026)

- RL over full harness

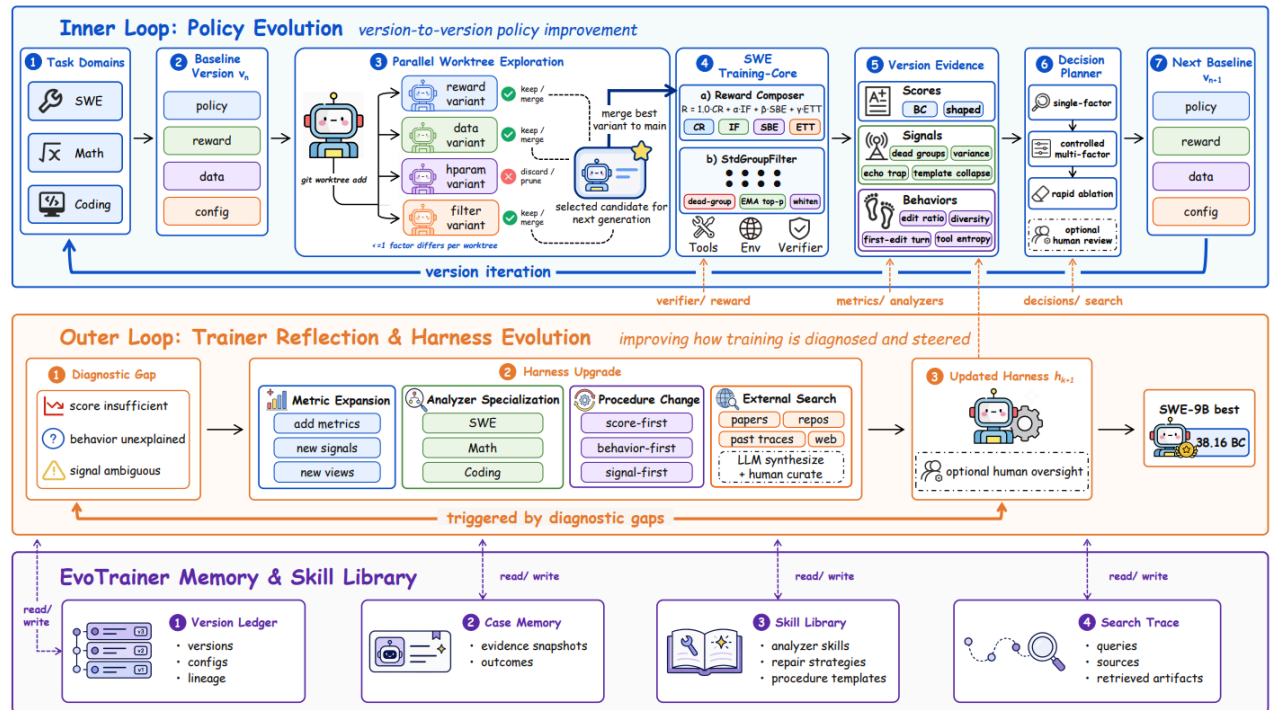
- train models holistically
- based on local outcomes

- Ideas:

- bootstrap using Foundation Models
- evaluate "local" and Foundation Models on same task
- can focus on difficult cases
- or large model fall-back

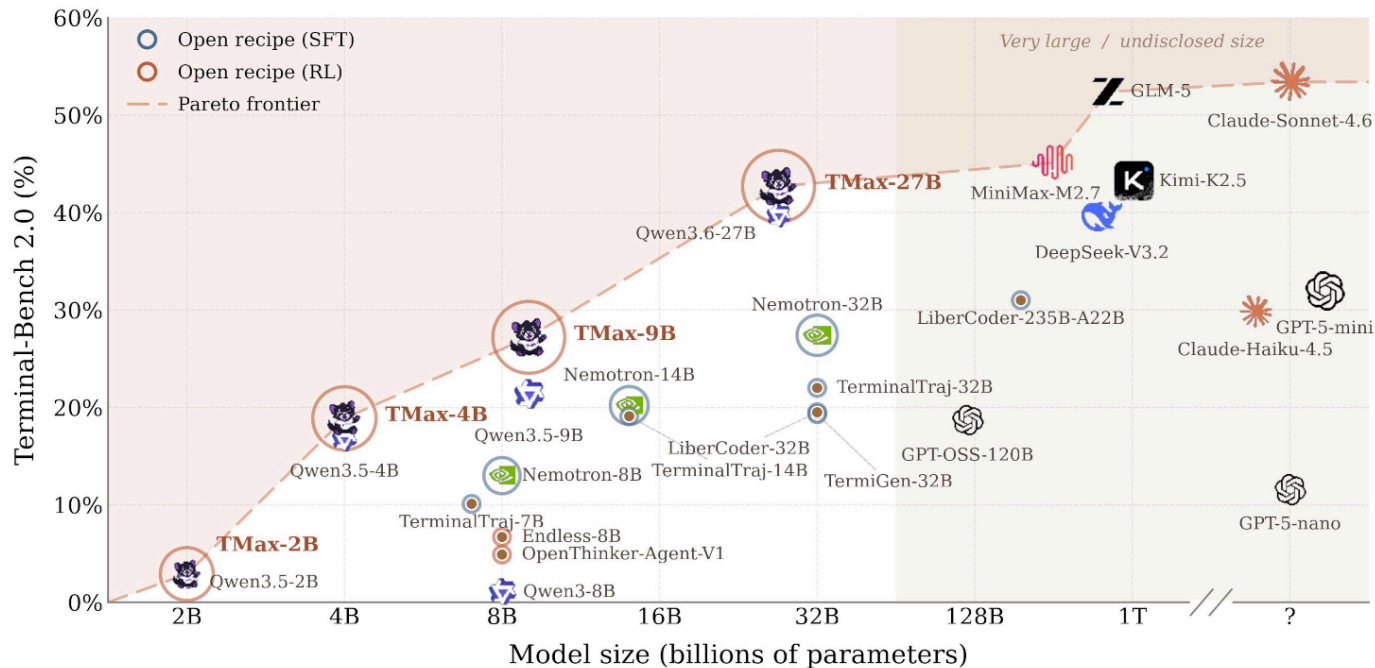
- Own your own AI

- rather than sending IP to SF



How feasible is RL inside the enterprise?

- TMax: A Simple Recipe for Terminal Agents + Apache 2.0 Code



Wrap-Up

- Harnesses can greatly enhance system performance
 - (even more so if you can specialise them)
- Recent geopolitical events should make everyone nervous ...
 - ... about relying on other people's AI
- Companies should also be thinking about Owing Their AI
 - and it's possible to implement this locally



Link to Slides

- https://bit.ly/MLSG_2026-06

