

A FIELD REPORT • SPRING 2026

The Agent *Ecosystem.*

A tour of the shape of the field — the metaphors we bring to it, the risks inside it, and the decisions everyone building is actually arguing about.

THE TERRITORY

The questions everyone building agents *is actually arguing about.*

01 Two metaphors

02 The lethal trifecta

03 Where agents live

04 Model vs harness

05 Performance vs quality

06 Managing complexity

07 Metaphors matter

08 — end of report —

Who is the agent *for*?

- AUDIENCE A

Hobby

(the family computer)

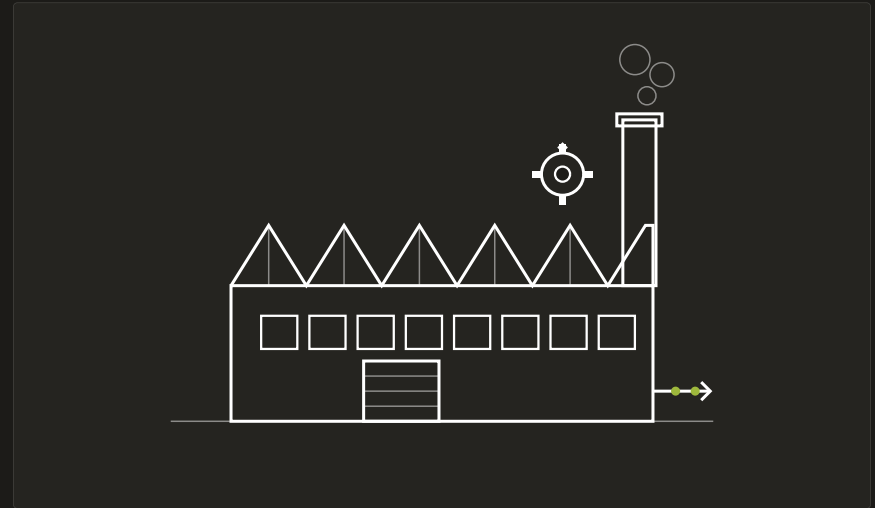


ONE TERMINAL, MANY USERS

- AUDIENCE B

Professional

(the factory)

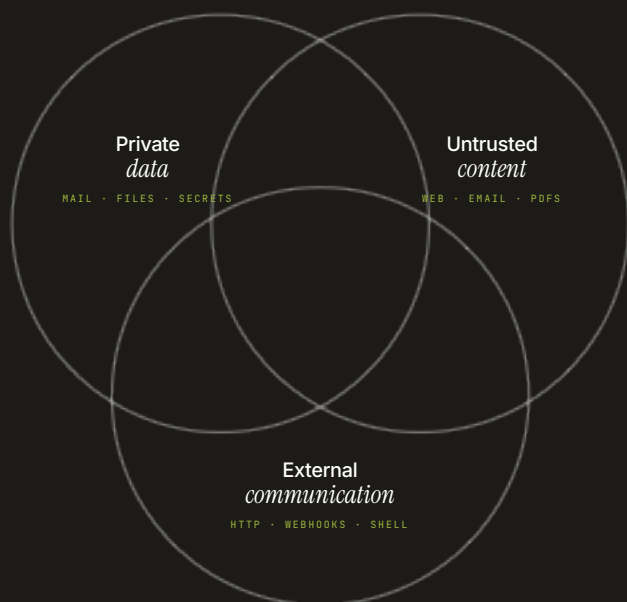


A SMALL FACTORY • FOR THROUGHPUT • MANY SEATS • SLAS

Two audiences. Two metaphors. Same models underneath — wildly different products.

The *lethal trifecta*.

Three capabilities that are each useful on their own. Combine all three in one agent and an attacker can drain your data with a sentence of text.

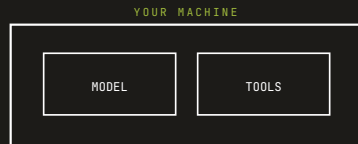


Where agents *live*.

Three zip codes. Each one gets you different trust, different latency, different blast radius.

A • ON METAL

Local hardware.



Yours end-to-end. Private by construction, limited by what fits in RAM.

OLLAMA • LM STUDIO • ON-DEVICE

B • CLOUD API

Cloud API.



Best capability, strongest guardrails — trust boundary at the TLS handshake.

FRONTIER APIS • MANAGED INFERENCE

C • SHIPPED

Binaries & software.



Agents as programs. Installable, sandboxed, carrying your creds.

CLI TOOLS • DESKTOP • IDE PLUGINS

Model *vs* Harness.

A • THE MODEL

What it *knows.*

- Reasoning, planning, judgment under ambiguity.
- Deciding *which* tool, *when*, *why*.
- Long-context coherence — staying on the thread.

vs

B • THE HARNESS

What it *can do.*

- Loop control, retries, budgets, interruption.
- Tool catalog, sandboxes, permissions, memory.
- Everything around the model that makes it *act*.

A smart model in a dumb harness acts stupid. A dumb model in a thoughtful harness gets real work done.

CHAPTER FIVE · COST & QUALITY

Performance *vs* Quality.

Not every step needs the biggest brain. Four principles for keeping agents fast, cheap, and correct.

PRINCIPLE 01

Match *context length* to the task.

PRINCIPLE 03

Bundle repeatable steps into *skills*.

PRINCIPLE 02

Use deterministic code for *80-90%* of the work.

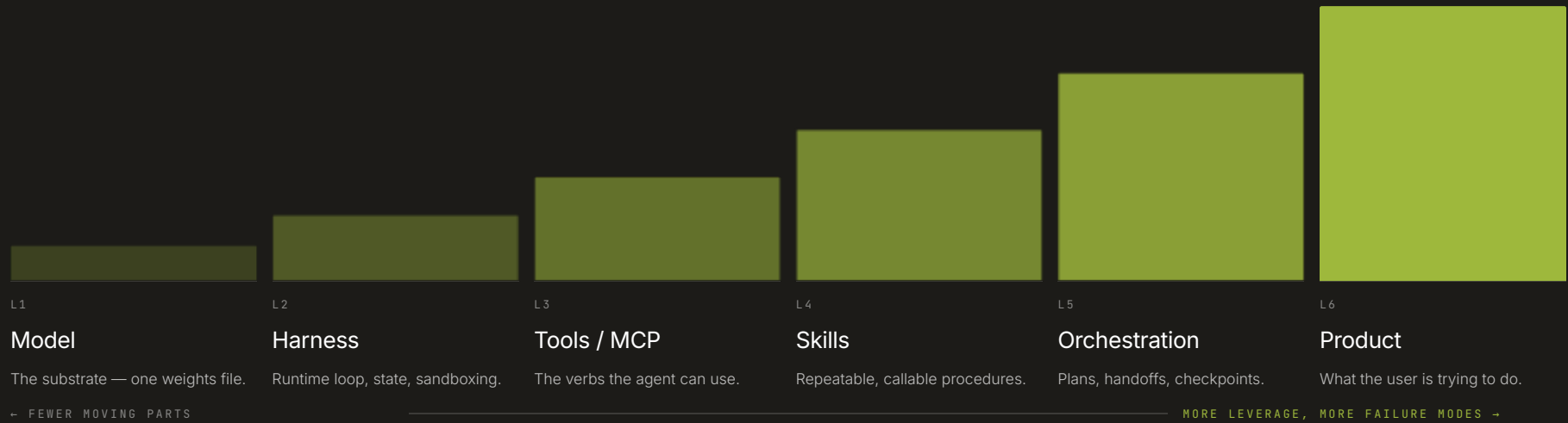
PRINCIPLE 04

Not every task requires *max intelligence*.

The goal isn't the smartest agent. It's the agent that gets this specific job done, reliably, for a price that makes sense.

Managing *complexity*.

Every layer you add is a layer that can fail. Every layer you remove is a layer you have to think about yourself.



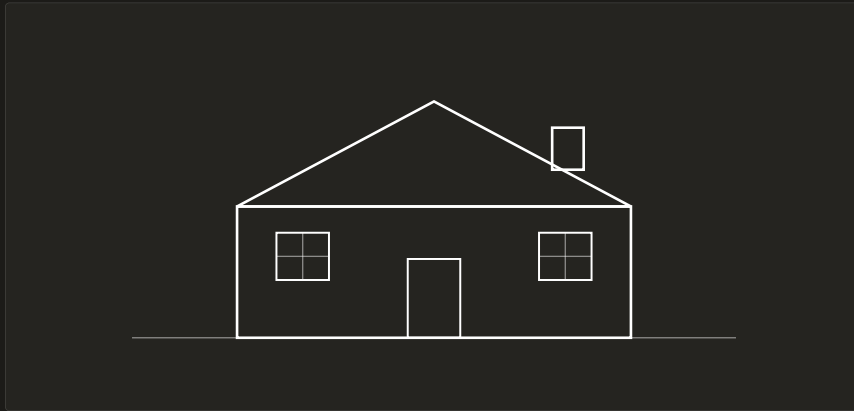
CHAPTER SEVEN • THE PUNCH LINE

Metaphors *matter*.

The metaphor you choose for the *work* — not the codebase, not the UI — shows up in every decision downstream.

METAPHOR A

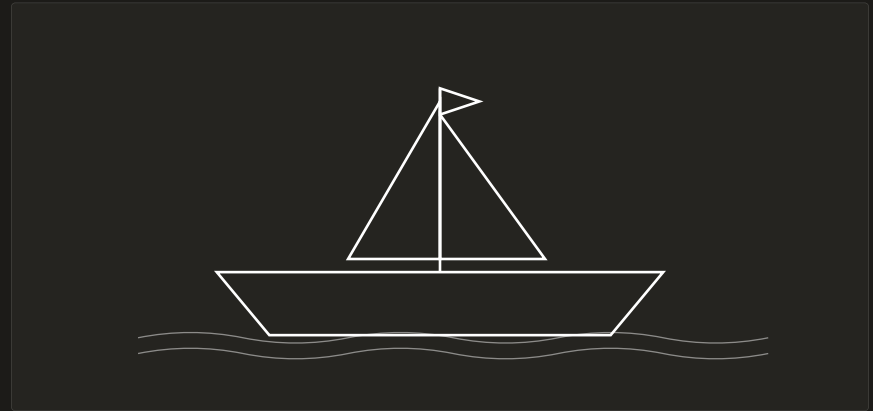
The work is a *house*.



SETTLED • ROOMS FOR PURPOSES • RENOVATIONS • OWNERS & GUESTS

METAPHOR B

The work is a *ship*.



IN MOTION • EVERY PART HAS A DUTY • CAPTAIN & CREW • HEADING SOMEWHERE

Same model, same team — different metaphor. You'll get different architectures, different review culture, different pagers at 3am.

CLOSING

Pick the *metaphor*.
Watch the *trifecta*.
Spend *intelligence* like money.

THANKS
– YOU, AUDIENCE

CONTINUED IN
PART 02

DATE
SPRING 2026

Q & A
YES PLEASE