When AI Leaves the Chat Window

Context, knowledge systems, CLI tools, and agents — how to become the most proficient user of AI in your organisation.

Barnaby Robson · Lakehouse Executive Training · February 2026

What you'll learn today

Six moves, from what AI can and can't do to building agents that work for you.

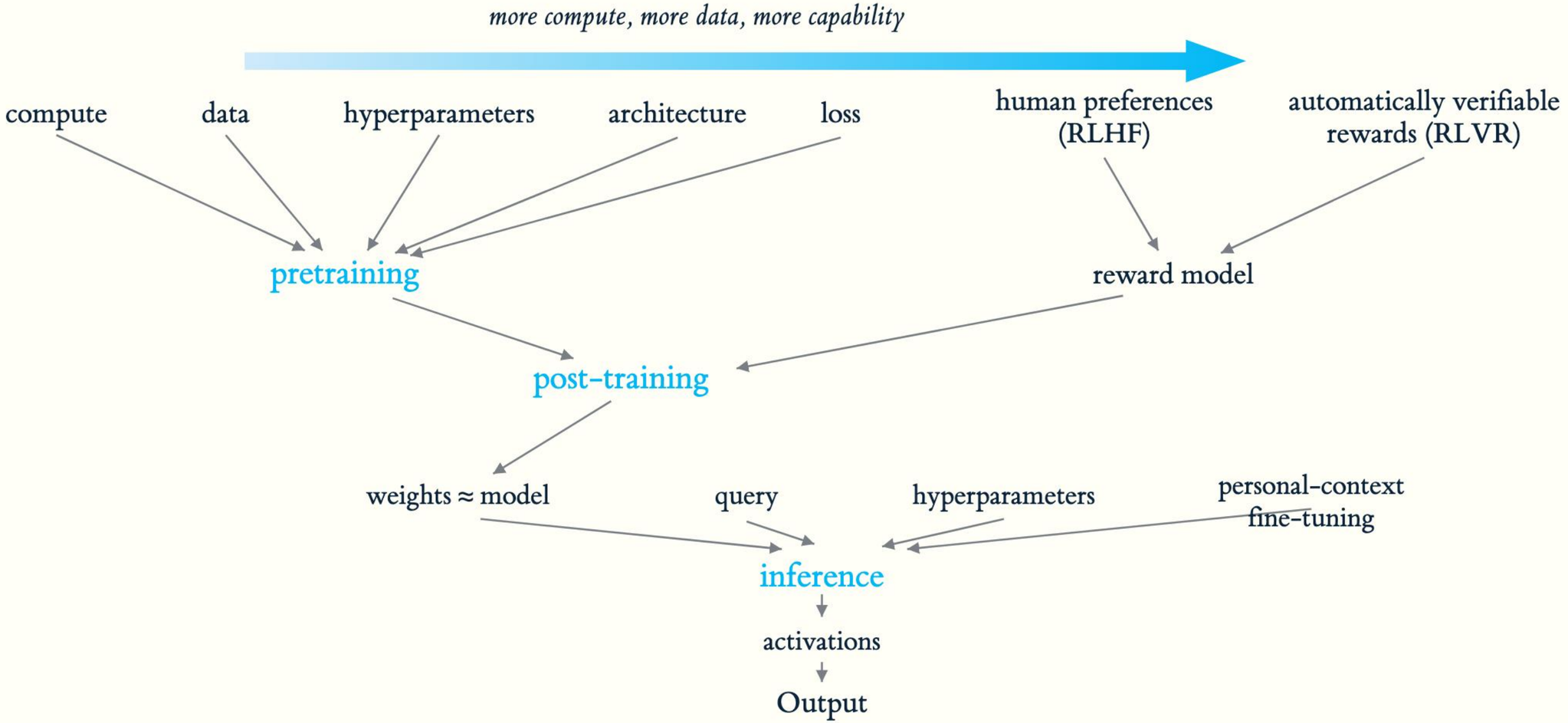
PART ONE

Understanding the basics

What these systems do remarkably well — and where they still fall down.

A bird's-eye view of machine learning

The arc runs from public knowledge at the top, through reinforcement, down to your own context at the moment of use.



After "The Scaling Era" (Dwarkesh Patel & Gavin Leech), adapted by Barnaby Robson. Success in 2026 comes from managing context at inference time.

A genius and a schoolchild in one system

Frontier models clear tasks that would take a person hours, and still trip on things a child would catch.

REMARKABLY WELL

What AI can do

- **Synthesise** information across vast domains
- **Draft** almost anything — reports, code, strategy
- **Translate** between languages and formats
- **Code** in any programming language
- **Analyse** documents, data and images
- **Reason** through complex, multi-step problems

NOT YET

What AI cannot do

- **Remember** across conversations, by default
- **Know your context** — company, history, preferences
- **Take responsibility** for a decision
- **Replace judgement** in ambiguous situations
- **Guarantee accuracy** — hallucinations remain real
- **Surface trade-offs** across a long context window

"We are not evolving animals, we are summoning ghosts." — Andrej Karpathy

The intelligence is spectacularly jagged

The same system swings between brilliance and basic error from one task to the next. Benchmark scores tell you little — test the tool on **your** work, in **your** context.

GENIUS POLYMATH

Clears the hardest tasks

- **Wins the International Maths Olympiad**
  gold-medal problems, worked end to end
- **Writes production-quality code**
  in any language or framework you name
- **Synthesises 100 documents in seconds**
  faster than any analyst could read them

CONFUSED SCHOOLCHILD

Trips on the trivial

- ✕ **Miscounts the letters in a word**
  the now-famous “strawberry” slip
- ✕ **Falls for a simple jailbreak prompt**
  a single line can redirect it
- ✕ **Forgets what you said five minutes ago**
  the context window has already moved on

“A genius polymath and a confused grade-schooler at the same time.” — Andrej Karpathy, on jagged intelligence.

Why AI hallucinates

A model predicts the next token under its training distribution. It optimises for sounding right, which is a different thing from being right.

WHAT IT DOES

Predicts the next token

- **Pattern-matches at scale**
  across the whole training distribution
- **Computes statistical probability**
  ranking which word most likely follows
- **“This sounds right, given the data”**
  fluency is the objective it optimises for

WHAT IT LEAVES OUT

The checks it skips

- ✕ **Grasping causation**
  correlation is all the text records
- ✕ **Verifying against reality**
  the corpus is the only ground it has
- ✕ **Asking “is this actually true?”**
  confidence and accuracy come apart

The rooster crows as the sun rises, so the model learns the correlation — while missing that the rooster never caused the dawn.

PART TWO

The importance of context

Frontier models accelerate early work, but their usefulness ends where continuity begins.

Every conversation starts from zero

We have all lost patience when the AI forgets an instruction we gave it five minutes before. Each session begins as if the system has never met your circumstances — which is effectively true.

No history

No project memory, no record of what was tried, no past preferences.

The same problem, at scale

Every time a key person leaves or a team rolls off, the context walks out the door.

Costly to move

The context exists. Moving it from a head into a system is the work.

Every firm has a “Rob Wall”

The person everyone gravitates to, because the context lives in their head. A project file keeps the answer; it rarely keeps the reasoning that produced it.

WHY

The reasoning behind decisions

The debate behind an assumption, the bend in the logic that never made it into the file.

WHO

Who can help with what

The map of relationships, internal and external, that quietly routes every request.

HISTORY

The history of stakeholder relationships

Why a target was passed on; where a negotiation stuck, and who softened it.

POWER

Who the real boss is

The unwritten structure no org chart shows, and whose nod actually unblocks a deal.

When Rob leaves, the insight leaves with him — sometimes to a competitor.

High intelligence is the ability to iterate, persist, and understand the big picture. The mark of low intelligence is the inability to learn from your mistakes.

— Dan Koe

So the questions for AI adoption are: how do we build systems that learn and retain, instead of starting from zero each time? And how do we keep more of the human context?

PART THREE

Where AI is going

The most important shift of the last year was a new interface — the agent that lives on your machine.

An agent that lives on your computer

The CLI — Claude Code and its kin — reads and writes your files, runs applications, and keeps context across sessions, all inside your own private environment.

MEMORY

Reads and writes files

Context persists between sessions.

SKILLS

Learns through markdown

New capabilities as plain text files.

ACCESS

Your files and context

Works on your world, on your data.

ACTION

Runs applications

It does the work, rather than describing it.

DAILY INSTALLS

17m

December 2025

31m

January 2026 — nearly doubled in a month

"Claude Code runs on your computer with your private environment, data and context." — Boris Cherny.

WHERE IT IS GOING

The hottest new programming language is English

You describe what you want; the AI turns intent into working code. Karpathy calls it **vibe coding** — you give in to the vibes and forget the code even exists.

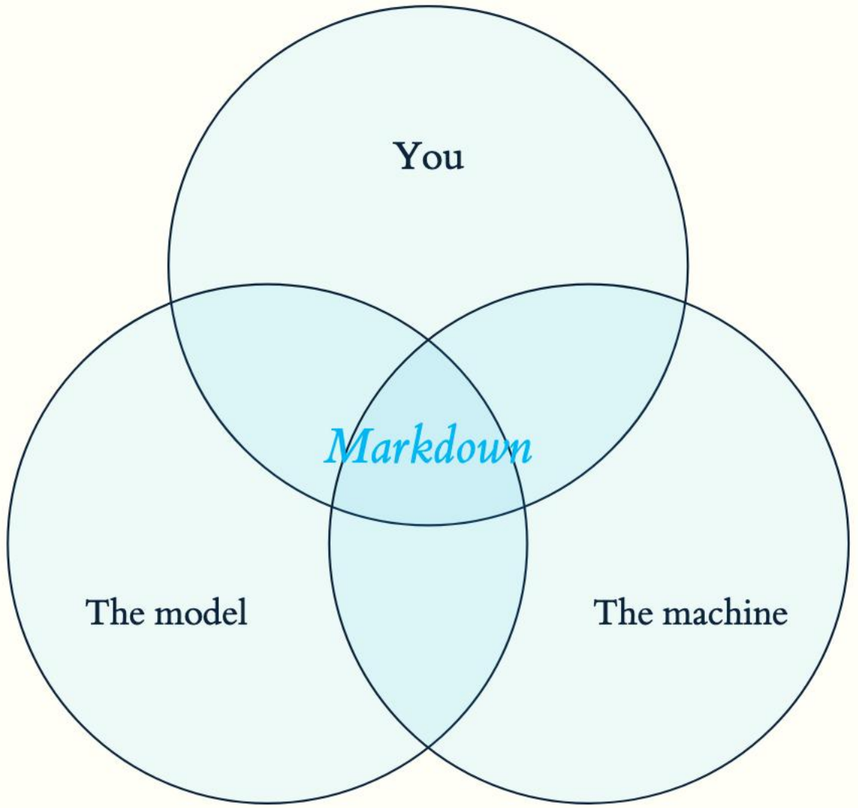
Building things now asks one thing of you: that you be clear about what you want.

— Andrej Karpathy

Someone wired Claude Code to their oven — no app, no integration, just plain language. 3.2m views.

Markdown is the shared language

One plain-text format a person can skim, a model can parse, and a machine can run. Everything you write in markdown becomes directly actionable.



"Everything you previously wrote in Markdown files can be directly actionable." — Kepano.

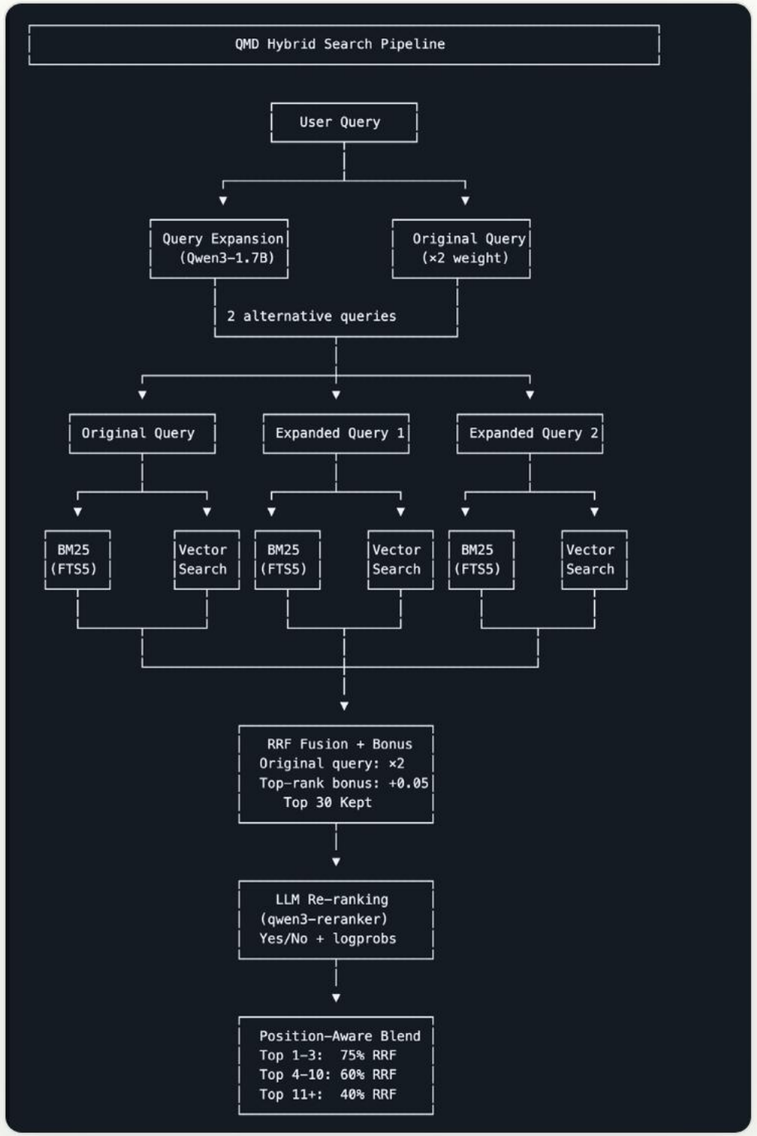
WHERE IT IS GOING

Small models, on your own device

A search engine for your knowledge base that runs entirely on your laptop. Notes never leave the machine; retrieval is by meaning, not keywords.

QMD, by Tobi Lütke, uses a 2GB model (Qwen 3) light enough for almost any device.

The direction of travel: capable AI that works entirely locally.



QMD's hybrid search pipeline — all running on-device.

The model moved from the cloud to your machine

The shift is about access to context — **your** context — held on your own machine, across time.

THE OLD MODEL

AI in the cloud

- **AI lives in the cloud**
  you visit it in a browser tab
- **Each conversation is isolated**
  nothing carries from one chat to the next
- **No access to your files**
  you paste the context in by hand
- **Starts blank every time**
  yesterday's work is already gone

THE NEW MODEL

AI on your machine

- **AI runs on your computer**
  inside your own private environment
- **Reaches your files, context and history**
  it works on your world directly
- **Persists and learns across sessions**
  memory accrues with every use
- **Becomes an assistant with memory**
  it remembers how you like to work

PART FOUR

How organisations must adapt

Knowledge work was priced on the cost of producing it. That cost just collapsed.

The economics of knowledge work have inverted

Clients still pay for judgement and for deciding what matters. Reassembling context that already exists somewhere in the firm becomes harder to charge for.

OLD MODEL — SCARCITY

Priced on effort

- ✕ **Output is costly to produce**
  hours of skilled time per deliverable
- ✕ **Cost stands in as a proxy for value**
  the bill tracks the effort behind it
- ✕ **Bigger problems mean more people**
  scale comes from adding headcount
- ✕ **The work is the value**
  the production itself is what you sell

NEW MODEL — ABUNDANCE

Priced on judgement

- **Production is cheap**
  a strong draft costs pennies and minutes
- **Context and judgement differentiate**
  what you know, and what you choose
- **Advantage goes to those who reuse**
  the firm that compounds its knowledge
- **The insight is the value**
  the judgement behind the work is what you sell

Modern firms risk occupying the position monasteries once did — valuable only because the work was slow and expensive.

After Treloar — the printing-press parallel

Before print, scribes were valuable because copying was scarce. Once copying went cheap, the value moved to authorship and interpretation. Are you a monastery, or a printing press?

Three responses that fail — all stuck in the chatbot paradigm

Each speeds up production or tightens control. None changes how judgement moves through the organisation.

RESPONSE ONE

Accelerate production

Deploy AI to generate faster. The work is saved, while the context is recreated each time.

CONTINUITY UNSOLVED

RESPONSE TWO

Wait for perfect systems

Delay until internal tools match every standard. Prudent, and still costly reconstruction while rivals cut marginal cost.

PRODUCTION UNCHANGED

RESPONSE THREE

Control without leverage

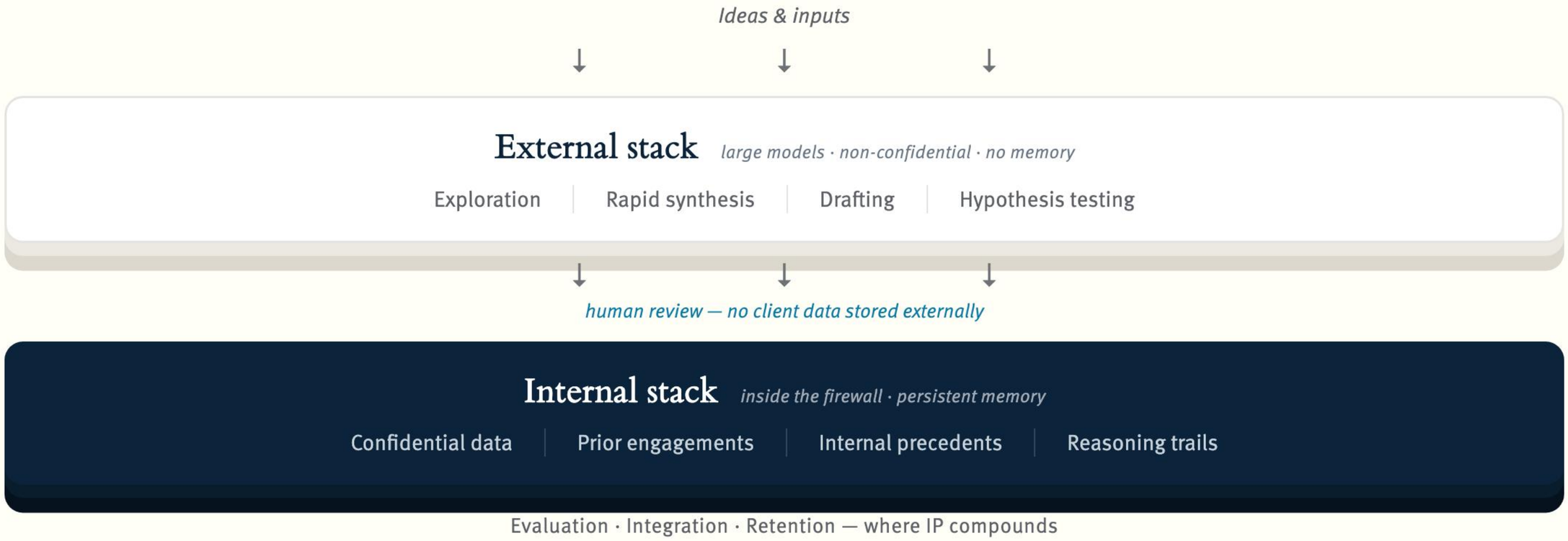
Keep tight controls but don’t adapt. You absorb costs competitors are eliminating.

COST ABSORBED

The same three dead ends appear across legal, banking and insurance — every knowledge-intensive industry.

The likely end-point: the two-stack firm

Winning firms split the work across two tiers: a large external model for fast exploration, and an internal model inside the firewall where confidential knowledge is retained and compounds. Everything flows down through a human-review gate before it joins the record.



How the two stacks work together

Explore on the outside, retain on the inside. Information flows inward, and nothing joins the record without a human reading it first.

01

Start local

Review internal precedents, prior work and stored knowledge first.

INSIDE

02

Go external

Reach for large external models only when internal knowledge runs out.

OUTSIDE

03

Human review

A person checks any external material before it becomes part of the record.

GATE

04

Feed back

Outputs and the reasoning behind them flow back to improve the systems.

RETAIN

05

Compound

Future teams inherit the knowledge without risking disclosure.

GROW

After Treloar — the two-stack firm.

Organise by meaning

Content legitimately belongs in several places at once. If the system can find it by meaning, where you filed it stops mattering.

TRADITIONAL — FOLDERS

Filed by location

- ✕ **One file, one place**
  though the content belongs in many
- ✕ **Rigid hierarchies**
  the tree decides where everything lives
- ✕ **Hard to find what you need**
  you must recall where you put it
- ✕ **Poor internal document search**
  keywords miss what you actually meant

BETTER — FILES OVER FOLDERS

Found by meaning

- **Files stay put; metadata does the work**
  categories, topics and links describe each note
- **Rich properties travel with the file**
  one note can sit in many views at once
- **AI retrieves on meaning**
  it finds by intent, wherever the file sits
- **Many paths to the same content**
  where you filed it stops mattering

The Kepano method, popularised by the founder of Obsidian.

The AID framework — a diagnostic for any value chain

Map your value chain. For each step, ask three questions and score it high, medium or low. Your AI strategy writes itself.

A · Automation

Can AI remove human labour? Document review, data entry, scheduling.
Map the repeatable steps and let the agent run them.
High-A ⋯⁘ efficiency to harvest.

I · Intelligence

Can AI make better decisions? Forecasting, pricing, risk scoring.
Feed the model your own data and let it score the call.
High-I ⋯⁘ advantage to capture.

D · Disruption

Does AI make this step obsolete? Intermediaries, gatekeepers, approvals.
Redesign the step before a rival removes it for you.
High-D ⋯⁘ existential risk to defend.

Defend the D risks, capture the I advantages, harvest the A efficiencies.

PART FIVE

Building agentic workflows

An agent is AI with tools, memory, and a method — it does the work, rather than describing it.

What an AI agent actually is

Chatting in a browser answers once. An agent runs the task — reading, writing, executing — and remembers.

Tools

Access to the file system, APIs and databases. **Reads and writes files on your computer.**

Action

Takes steps — read, write, execute. **Runs code and connects to external services.**

Memory

Persists across sessions. **Carries context forward instead of starting blank.**

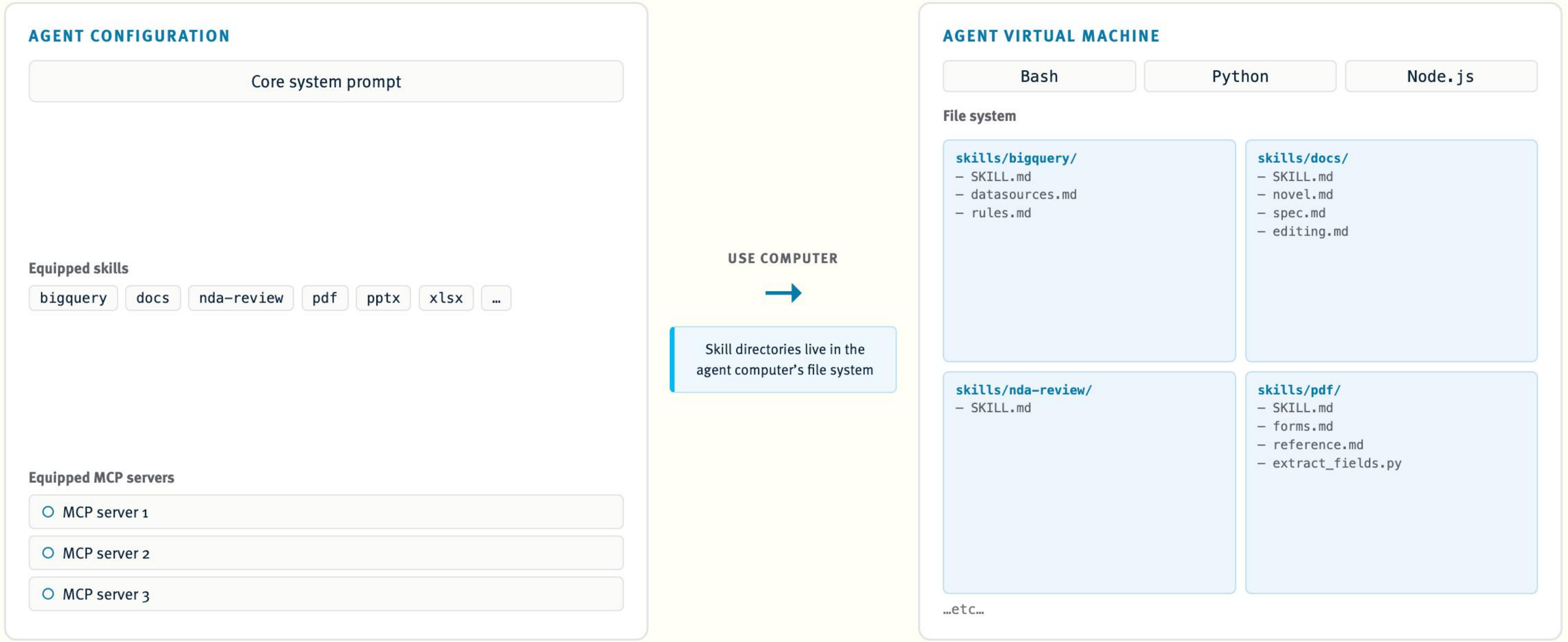
Skills

Custom capabilities for specific tasks. **Learns your preferences and your methodology.**

Claude Code is the worked example throughout this section.

Agents, skills, MCP, and virtual machines

A skill is a simple text file that tells the agent how to do one job — when to act, what to do, which tools to use, how to format the result. Skills, MCP connectors and a sandboxed machine are the building blocks.



The agent runs in its own virtual machine; equipped skills become directories in its file system. After Anthropic.

A skill is a plain-text file the agent loads when it's needed

Write a procedure once — the method, the steps, the standard — and every agent runs it the same way. Here is one from deal advisory.

```
synergy-reviewer/
├─ SKILL.md          ◀ the method — the "very little risk" standard, the four-attributes test, in plain markdown
├─ scripts/          ◀ the deterministic parts — a tested script for steps where improvising would slip
│  └─ attribution.py
└─ references/       ◀ deep knowledge — structural vs operational barriers, pulled in only when the work calls for it
   └─ barriers.md
```

```
description: "use when reviewing synergy assumptions for deliverability and
transaction attribution"
```

▲ one line decides when the skill loads, so the context window stays clear until it's needed

The agent follows your methodology — EDGE 3.0, FDD red flags, Day-1 checklists — rather than generating generic content.

AGENTIC WORKFLOWS

The system learns from its own mistakes

Correct the agent once. A “reflect” skill finds the root cause, proposes a specific fix to the skill or config, and — with your approval — writes it back so the error doesn’t return.

Compound engineering — the agent ships and learns while you sleep.

```
┌─ Skill Reflection: reflect + Global Configuration ──────────────┐
│                                                                 │
│  Signals: 1 correction (HIGH), 2 successes (MED), 1 coverage gap │
│                                                                 │
│  CORRECTION DETECTED:                                           │
│  User: "the reflect skill is here - can you use the skill to    │
│         reflect on the fact that you don't seem to know where   │
│         your skills are"                                        │
│                                                                 │
│  ROOT CAUSE: Searched ~/.claude/skills/ (doesn't exist) instead │
│  of following Step 1 of reflect workflow:                       │
│  "Check user's CLAUDE.md for declared skill locations"          │
│                                                                 │
│  The Global CLAUDE.md at line 7-8 explicitly states:            │
│  "Custom skills are stored in:                                  │
│   ~/Library/Mobile Documents/.../claude-code/skills/"           │
│                                                                 │
│  Proposed changes:                                              │
│                                                                 │
│  🔴 [HIGH] reflect/SKILL.md - Make "Check CLAUDE.md"            │
│     MANDATORY FIRST step before any directory scanning          │
│                                                                 │
│  🔴 [HIGH] Global CLAUDE.md - Add explicit instruction:         │
│     "ALWAYS read this file first when looking for skills"       │
│                                                                 │
│  Apply these changes? [Y/n]                                     │
└─────────────────────────────────────────────────────────────────┘
```

A real reflect run — correction, root cause, proposed fixes, awaiting your ‘yes’.

Your notes become fuel for creation

Because the agent can see your context, ordinary inputs turn into finished artefacts — work that would otherwise take hours of assembly.

Discuss a problem

⇢ a structured task list — the steps, in order, ready to act on.

Research a topic

⇢ connected ideas, linked across your vault and traced back to source.

Review your notes

⇢ the patterns you had missed, surfaced and named.

Plan a project

⇢ a full project plan, drafted from your own material.

The input is what you already capture; the output is something you can use.

Agency is significantly more powerful, and significantly more scarce, than intelligence.

— Andrej Karpathy

Agency is the capacity to take initiative, make decisions, and exert control over your actions. In an AI-abundant world the differentiator is the willingness to act. Are you hiring for agency? Educating for it? Acting as if you had 10× agency?

PART SIX

Personal effectiveness

What this looks like when one person, with no coding background,

just starts.

From simple tasks to a knowledge base rebuilt in one weekend

First weekend with Claude Code — restructuring 10,000 notes, while out hiking.

- Re-structured **10,000 notes** on file-over-folder logic
- Optimised the website's SEO
- Created new skills for specific workflows
- Ran several agents in parallel — while hiking

THE WEEKEND

10,000+
files imported from Notion

8,000+
files with metadata added

~£200
total cost in AI tokens

How this works in an ordinary day

A single loop, repeated: capture, organise, ask, create — and the system gets a little better each time.

01

Capture

Notes and ideas land in Obsidian, in plain markdown.

YOU

02

Organise

The AI categorises and links them automatically.

AGENT

03

Ask

Put a question; the agent searches the whole vault.

YOU

04

Create

Produce content from accumulated knowledge.

AGENT

05

Improve

Skills sharpen through use, so tomorrow is faster.

COMPOUND

Simple, high-impact prompts

These work because the AI has **your** context — the prompts themselves are plain.

RESEARCH & IDEATION

"Review my vault and suggest topics I should write about, based on what I've been capturing."

⇢ a shortlist of angles drawn from what you already return to.

PEOPLE ENRICHMENT

"Scan my notes to identify the people mentioned, and create an Authors & Quotes section."

⇢ a linked directory of who you cite, with their sharpest lines.

NOTES TO PRESENTATIONS

"Turn my prep notes into a presentation, using this formatting."

⇢ a formatted deck, built straight from your prep — this one began that way.

PATTERN FINDING

"Find patterns across my notes on this topic. What connections exist?"

⇢ the threads tying together notes you wrote months apart.

Each works because the agent reads YOUR context — the replies are specific, never generic.

Your call to action

The only way to learn is by doing. Start small, this week.

THIS WEEK

Install & play

Obsidian (free) and Claude Code ($20/month). Spend two hours just playing.

THIS FORTNIGHT

Build your system

A small personal knowledge base. Capture notes and meeting takeaways in markdown.

THIS MONTH

Write a skill

One custom skill for something you do repeatedly — a method, a regular process.

ONGOING

Raise your agency

Ask: what would I do with 10× agency? Then use the tools to act on the answer.

THE PROVOCATION

Every one of you could be in the top 0.1% of AI proficiency — within a fortnight.

The technology is ready, the tools are available, and the AI itself can teach you. The only question is whether you act.

Barnaby Robson · barnabyrobson.org