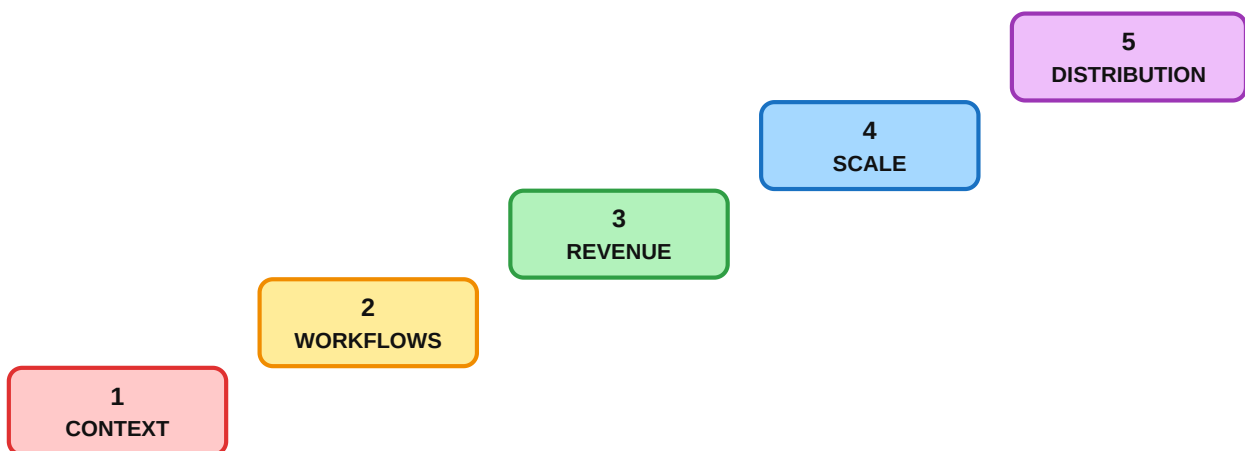


# 5 LEVELS TO GET RICH WITH CLAUDE

Everything from the video in one place. The whiteboard, the prompts, the setup, and one master prompt that turns this PDF into your own operator.



## Miles Deutscher

Founder, AI Edge. 3 years of daily AI. A 30-person company running on this exact system.

## Read this first

I run a 30-person company and Claude sits in every layer of it. Not as a chatbot. As infrastructure.

There are five levels to making money with AI. Most people never leave Level 1, and it is not their fault. Nobody shows them the ladder. This guide is the whole ladder: what each level is, how the money shows up, the exact diagrams from the video, the setup, and the prompts.

Two ways to use it. Read it top to bottom once so you can see the whole machine. Then go to the last page, copy the master prompt, attach this PDF to your own Claude, and let it rebuild the entire playbook around your business. That page is why this document exists.

---

### HOW THE MONEY SHOWS UP AT EACH LEVEL

| LEVEL | NAME                | WHAT YOU BUILD                       | MONEY =                |
|-------|---------------------|--------------------------------------|------------------------|
| ● 1   | <b>Context</b>      | AI that actually knows your business | <b>Decisions</b>       |
| ● 2   | <b>Workflows</b>    | Automate everything that repeats     | <b>Time</b>            |
| ● 3   | <b>Revenue</b>      | Point the machine at money           | <b>Income</b>          |
| ● 4   | <b>Scale</b>        | Build once, sell many                | <b>Margin + equity</b> |
| ● 5   | <b>Distribution</b> | Own the pipe, AI runs it             | <b>Compounding</b>     |

Everything is additive. Each level runs on the one below it. The folder you build at Level 1 fuels the entire machine, which is why skipping it is the most expensive mistake on this list. And there is a twist at Level 5 that changes what the whole thing actually is. It's on the second to last diagram.

**You know exactly which level you're on right now. The only mistake is staying there.**

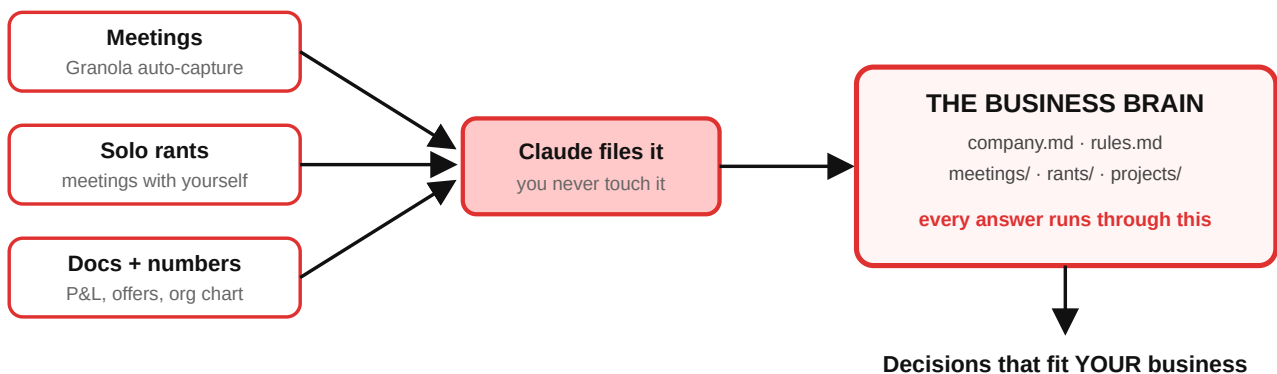
## AI that knows your business

**Money = decisions.** Level 1 is the one everyone skips, and it is why their AI gives generic advice. ChatGPT gives everyone the same answer because it knows nothing about you. The fix is not better prompts. It's memory.

I built a folder that knows my entire company. Numbers, offers, team, deals, the way I make decisions. Every meeting gets auto-captured with Granola and Claude files it into the folder. I never touch it.

No meetings? Have meetings with yourself. I record solo rants, unstructured voice notes about what's on my mind in the business, and Claude files those too. Ten minutes of rambling becomes structured context forever.

The payoff is the side-by-side. Ask a cold Claude "what's my biggest risk right now" and you get a horoscope. Ask the same question inside the folder and it names the client, the number, and the deadline. Same model. Different level.



The Level 1 memory system. Capture is automatic, filing is Claude's job, the folder is the asset.

**The model is rented. The memory is yours. That's the moat.**

## Build your business brain this week

### THE FOLDER SKELETON

```
business-brain/
├─ company.md      what the business is, offers, real numbers
├─ rules.md        how you make decisions, what you say no to
├─ meetings/       Granola drops transcripts here
├─ rants/          your solo voice notes, filed
├─ projects/       one file per active project
└─ decisions.md    a running log of every call you make, and why
```

### PROMPT 1 · BUILD THE FOLDER (PASTE INTO CLAUDE CODE, INSIDE AN EMPTY FOLDER)

Set up my business brain. Interview me one question at a time until you can write a complete company.md (what we do, offers, pricing, team, real numbers) and rules.md (how I make decisions, what I always say no to, my risk limits). Then create meetings/, rants/, projects/ and decisions.md. Keep every file short, factual and current. When something I say contradicts an existing file, update the file and tell me what changed.

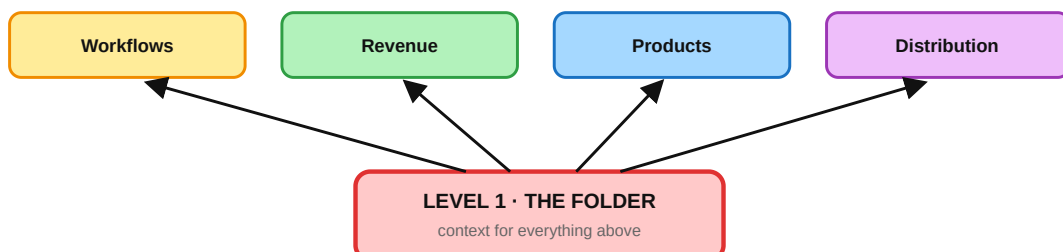
### PROMPT 2 · THE FILING HABIT (RUN AFTER EVERY MEETING OR RANT)

New transcript in meetings/ (or rants/). File it: pull out decisions, numbers, commitments and risks, update the relevant project file and decisions.md, and flag anything that contradicts rules.md. Give me a 5-line digest, nothing else.

### PROMPT 3 · THE TEST (THIS IS THE WHOLE LEVEL IN ONE QUESTION)

Based on everything in this folder: what is my biggest risk right now, what would you do about it this week, and what am I not seeing? Name names and numbers. No generic advice.

Run Prompt 3 in a fresh Claude chat with no folder, then inside your business brain. The gap between those two answers is what Level 1 is worth.



The flagship system view. One folder underneath, four money layers on top of it.

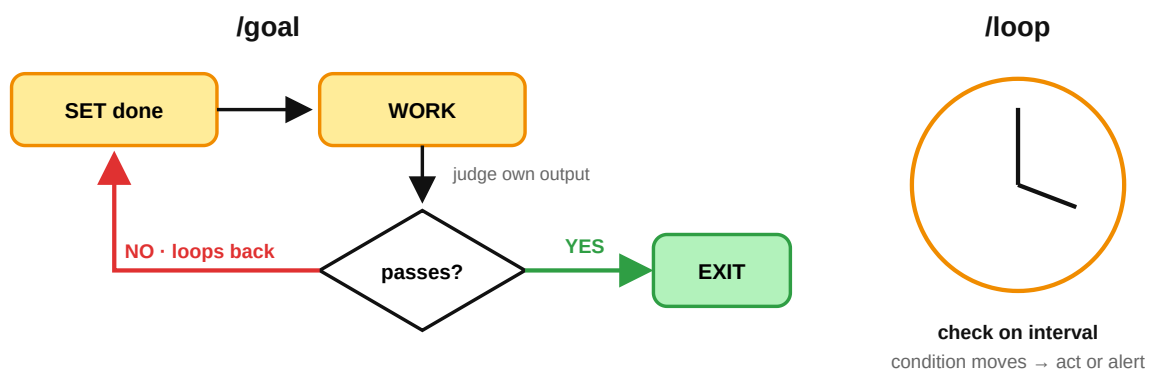
## Automate what repeats

**Money = time.** Level 2 starts with an audit, not a tool. Open a spreadsheet and list every task you did this week: what it was, minutes it took, how often it repeats. Anything that repeats with the same shape is a workflow, and workflows are what Claude eats.

Then there are exactly two commands you need, and one decision line between them:

- **/goal** is for pushing to a finish line. You define done, Claude works, judges its own output against the definition, and only exits when it passes. You are not in the loop. That's the point.
- **/loop** is for watching for change. It runs on an interval, checks a condition, and acts or alerts when something moves. Competitor uploads, inbox triage, a metric crossing a line.

Pushing to a finish line? /goal. Watching for change? /loop. That one sentence sorts every task on your audit sheet.



Pushing to a finish line? /goal · Watching for change? /loop

The Level 2 diagram. The red arrow is the aha: Claude judges its own work and loops back without you.

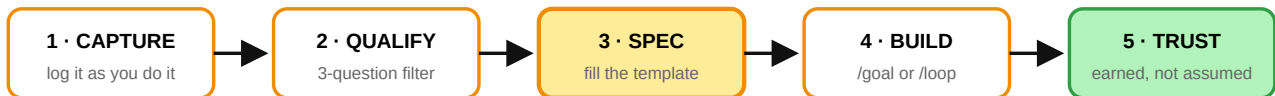
PROMPT · THE TASK AUDIT (PASTE INTO YOUR BUSINESS BRAIN)

Build me a task audit. Interview me about everything I did in the last 7 days. Put it in a table: task, minutes, times per week, same shape every time (y/n). Then mark each row /goal, /loop, or keep-human, and rank by minutes saved per week. Give me the top 3 to automate first.

**You stop being the loop. You start designing the loop.**

## Turn any task into an automated task

This is the repeatable system. Every task on your audit sheet goes through the same five steps, and the spec template in the middle is what makes the automation actually work. A vague task makes a vague agent. A filled spec makes an employee.



**Step 2, the filter.** Three questions: Does it repeat with the same shape? Are the inputs digital? Can you define done in one sentence? Three yeses means automate it. Any no means keep it human for now and move to the next row on your sheet.

### STEP 3 · THE SPEC TEMPLATE (FILL EVERY BRACKET, SAVE AS A FILE IN YOUR BUSINESS BRAIN)

```

TASK: [name it like a job title, e.g. "inbound lead qualifier"]
TRIGGER: [when it runs: on demand / every morning / when X changes]
INPUT: [exactly what it reads: file, inbox, folder, URL]
RULES: [which parts of rules.md apply, plus any task-specific rules]
STEPS: [the 3-6 moves you make when YOU do it manually]
OUTPUT: [what done looks like, one sentence, concrete]
CHECK: [how Claude judges its own work before showing you]
ESCALATE: [what it must NEVER do alone: send, spend, delete, promise]
  
```

### STEP 4 · THE UNIVERSAL CONVERSION PROMPT (PASTE SPEC, RUN INSIDE YOUR FOLDER)

```

Here is a task spec. First tell me: /goal or /loop, and why, using the decision line
(finish line vs watching for change). Then build it: write the exact command I will run,
test it once on the sample input, judge the output against CHECK, and fix it until it
passes. Respect ESCALATE absolutely: draft, never send. Show me the result plus a 3-line
note on what you would improve after 5 runs.
  
```

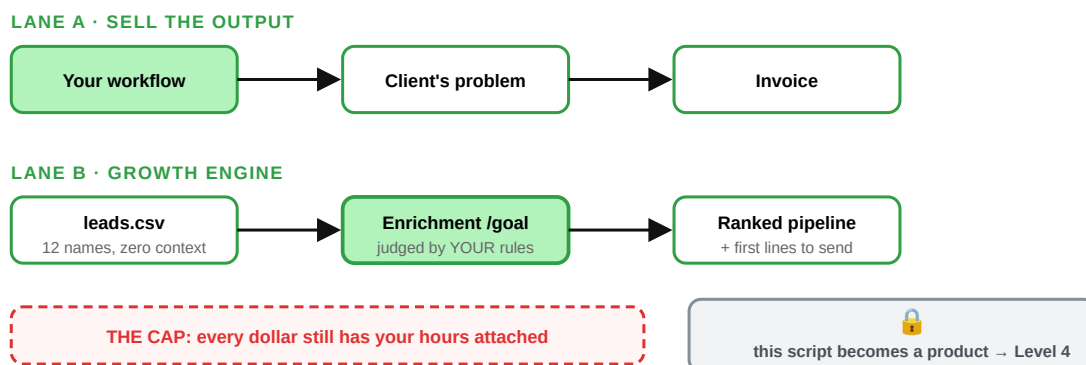
**Step 5, the trust ladder.** Run 1: watch everything it does. Runs 2 to 4: review the output only. Run 5 onward: spot-check weekly. Anything in ESCALATE stays yours forever. Automation is speed you earn, one clean run at a time.

## Point the machine at money

**Money = income.** Most people never make a dollar from AI because they stop at Level 2 and optimize their own life forever. Level 3 is pointing the workflows at things people pay for. Two lanes:

- **Lane A, sell the output.** The employee freelances the exact skill they automated at work. The freelancer triples throughput at the same rates.
- **Lane B, build growth engines.** If you have a business, this is where BD unlocks. Lead gen, ad analysis, outbound at scale. Same pattern: a workflow, pointed at revenue.

The demo from the video: a prospect list, twelve names, zero context. The Level 2 engine enriches it against my rules from the Level 1 folder. It is not filling columns, it is making judgment calls: it scores every lead, rejects bad fits, even declines money for the right reasons, and drafts first lines I can actually send. Boring list in, ranked pipeline out. That's a \$2K a month tool subscription running as a script on your machine.



The two revenue lanes, the honest cap in red, and the padlock that pushes you up the ladder.

**PROMPT · THE ENRICHMENT ENGINE (RUN INSIDE YOUR BUSINESS BRAIN, SWAP IN YOUR CSV)**

```
/goal Enrich leads.csv. For each lead: research who they are, score fit 1-10 against rules.md, flag conflicts with any current client or rule, and reject anything below 7 with one line explaining why. For everything 7+, write a first outreach line in my voice using context from this folder. Done = a ranked enriched-leads.csv plus a 5-line summary of who to contact first and why.
```

The honest part: Level 3 money still needs your hours. Fastest money on the ladder, weeks, no product, no audience. But it stops when you stop, and that cap is what pushes you to Level 4.

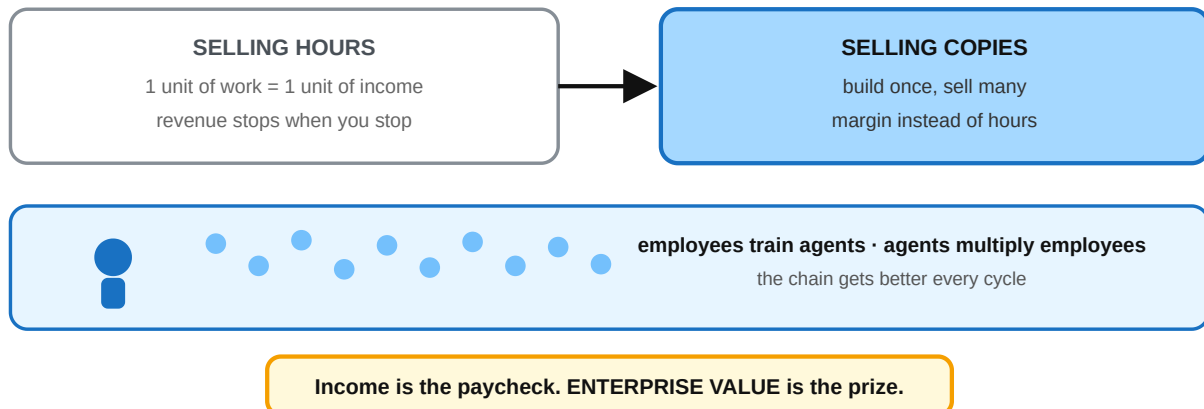
## Build once, sell many

**Money = margin and equity.** Remember the padlock? The enrichment script from Level 3 is a service while you run it for clients. Wrap it so they can run it themselves and it becomes a product. Build once, sell many. Margin replaces hours, and for the first time on the ladder your income is no longer a function of your calendar.

Inside a company, the same physics show up as economies of scale. My people do not just use agents. Employees train agents, agents multiply employees, and the chain gets better every cycle. One operator with ten trained agents outships a department, and every correction they make compounds into the system.

And this is where the frame shifts from income to worth. Income is what the machine pays you. Enterprise value is what the machine is worth. Systems that run without you are the difference between owning a job and owning an asset someone would pay a multiple for.

### ECONOMIES OF SCALE



Level 4: from hours to copies, one person multiplied by ten agents, and the value line in gold.

### PROMPT · PRODUCTIZE A WORKFLOW

```
/goal Take my [workflow name] from Level 3 and turn it into something a stranger could run without me. Done = a tool with a simple interface, my rules baked in as defaults they can edit, a 1-page setup guide, and a pricing suggestion based on the time and money it saves. Ask me a maximum of 5 questions first, then build.
```

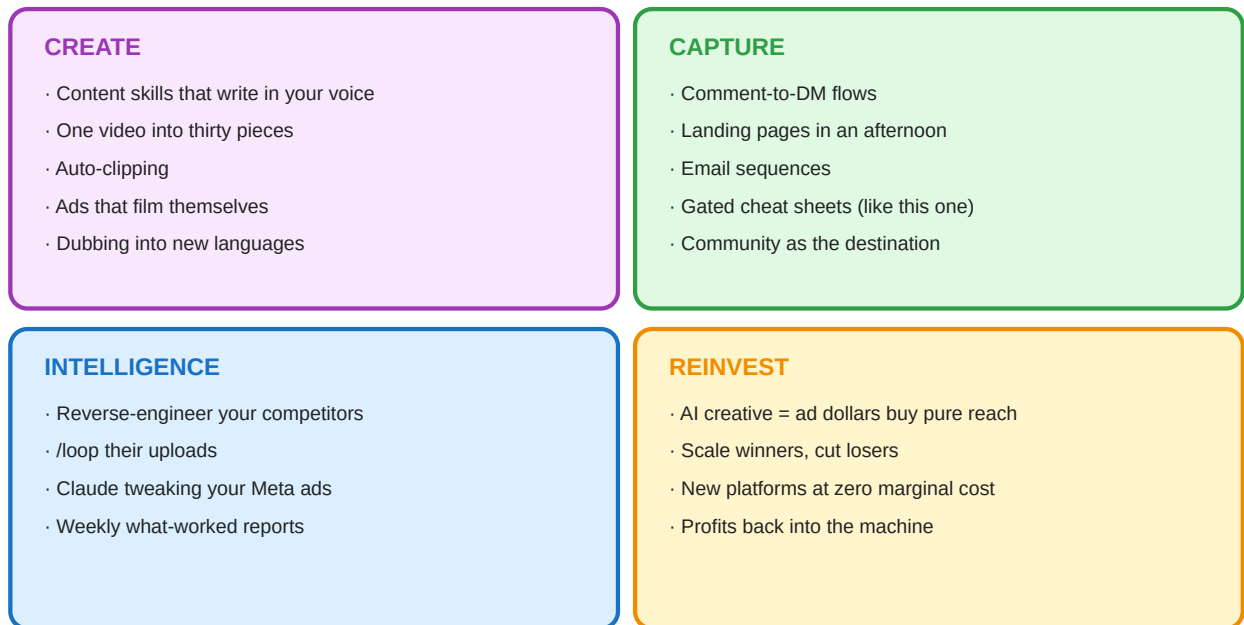


## Own the pipe

**Money = compounding.** Everything below this level assumes someone shows up to buy. Distribution is making sure they do, on channels you own. Posting is renting. Capturing is owning. Build the capture system before you expect anything to sell.

Everything AI can run on this wheel falls into four buckets. Every box on this board was somebody's full-time job three years ago. Don't do all of it. Pick three boxes and ship them this week.

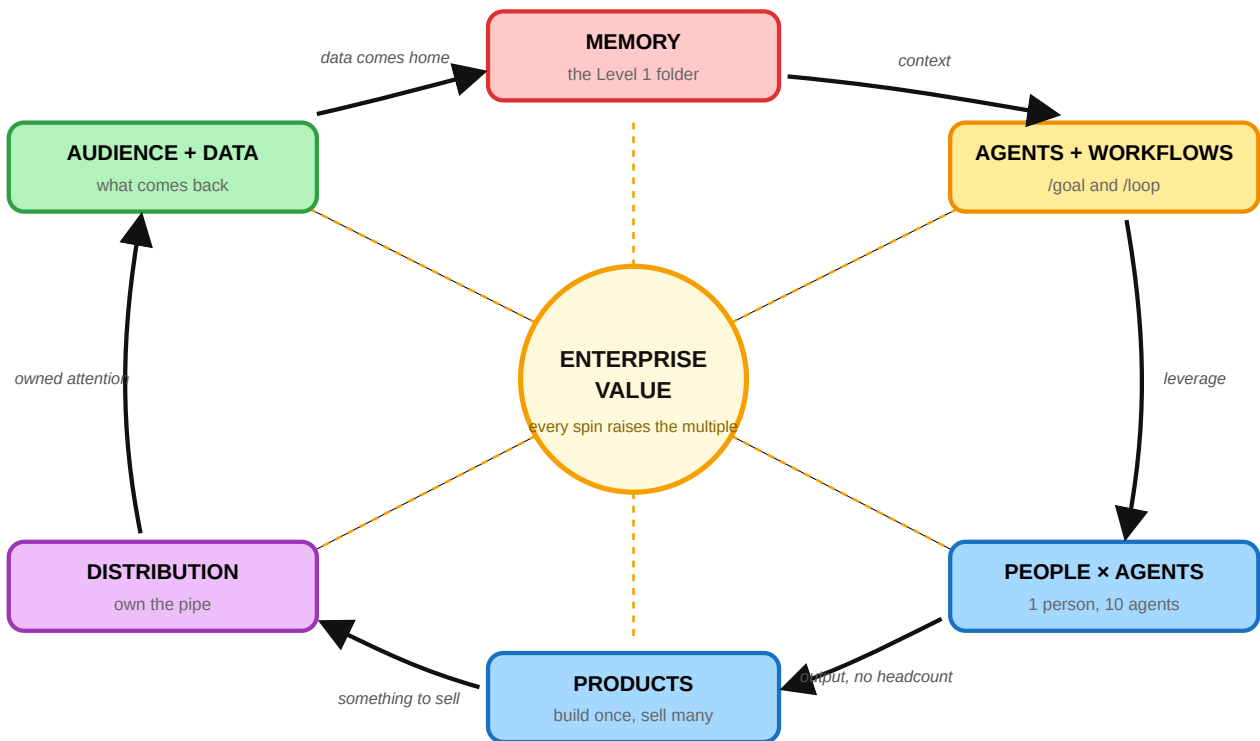
### THE DISTRIBUTION PLAYBOOK



Four buckets, scan don't read. Pick three boxes. Ship them this week.

## It was never a ladder

Here's the twist the whole video builds to. Level 5 feeds Level 1. Your audience and your data come home to the memory, the memory makes the agents smarter, the agents make the products better, the products grow the distribution. It's a flywheel, and the machine itself is the asset.

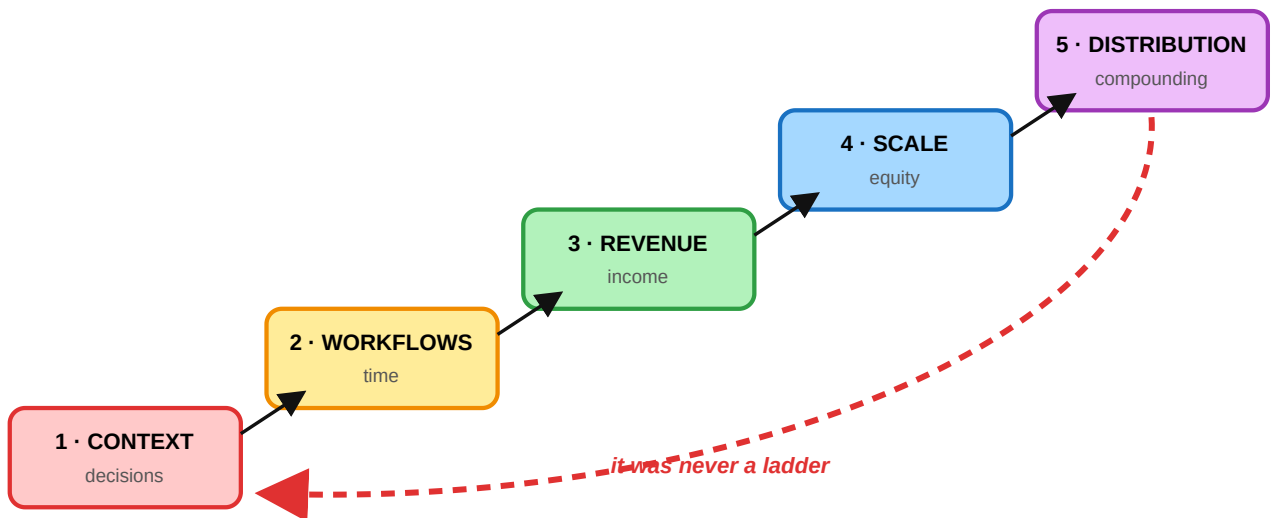


The flywheel. Six nodes, one machine, enterprise value in the middle collecting from every spin.

**Income is what the machine pays you. Enterprise value is what the machine is worth.**

## Five levels, one machine

Context buys you decisions. Workflows buy you time. Revenue buys you income. Scale buys you equity. Distribution buys you compounding. And the red arrow makes it one thing: Level 5 feeds Level 1, forever.



The final board from the video: the full staircase with the red loop back.

## WHERE TO START, BY SITUATION

- **Employee:** Level 1 on your role, Level 2 on your busywork, Level 3 by freelancing the skill you just automated.
- **Freelancer:** Level 2 on delivery to triple throughput at the same rates, then Level 4 to productize.
- **Business owner:** Level 1 this week, no exceptions. Then Lane B growth engines at Level 3, and the agent org at Level 4.
- **Creator:** You live at Level 5. Your gap is almost always Level 1 and the capture column.

## Setup: get running in 20 minutes

### 1 · WHAT YOU NEED

A Claude plan at **claude.ai** (Claude and Claude Code both come with it), plus **Granola** if you want meetings auto-captured. That's the whole stack. No \$2K a month tools.

### 2 · CLAUDE CODE (THE ENGINE FOR /GOAL AND /LOOP)

Install it, then open it inside your business-brain folder so every command runs with your context:

```
npm install -g @anthropic-ai/claude-code
cd business-brain
claude
```

Full docs and platform requirements: [docs.claude.com](https://docs.claude.com) (Claude Code section).

### 3 · GRANOLA MCP (AUTO-CAPTURE YOUR MEETINGS)

In [claude.ai](https://claude.ai) go to **Settings** → **Connectors** → **Add custom connector** and paste **<https://mcp.granola.ai/mcp>**, then sign in when the OAuth window pops. From there Claude can pull any meeting note straight into your folder. No meetings? Record solo rants in Granola and file those instead.

### 4 · THE RHYTHM THAT MAKES IT COMPOUND

**Daily:** capture (Granola does it for you). **Weekly:** run the filing prompt, then the risk question.

**Monthly:** re-run the task audit and automate the new top 3. The folder gets smarter every cycle, and everything built on top of it inherits that.

One honest note on safety: keep real credentials and client-confidential material out of the folder, give agents paper-trading or draft-only permissions before anything touches money or gets sent, and read what /loop wants to do before you let it act. Speed is the reward for trust earned, not assumed.

## The master prompt

Attach this PDF to a new Claude chat (or drop it in your business-brain folder), paste the prompt below, and Claude will reverse engineer the entire playbook around your situation. This turns a guide you read once into an operator you work with daily.

You are my AI operator. I have attached the "5 Levels to Get Rich with Claude" playbook. Read it end to end before you say anything.

Then work through this, in order, one step at a time:

1. INTERVIEW ME. One question at a time, maximum 8 questions. Figure out: what I do for money, what my week actually looks like, whether I have a business / clients / an audience, and what tools I already use.
2. DIAGNOSE MY LEVEL. Tell me which of the 5 levels I am really at (not where I think I am), and what the single most expensive gap is. Be blunt.
3. BUILD MY LEVEL 1. Write my company.md and rules.md drafts from the interview, propose my folder skeleton, and tell me exactly what to capture this week (meetings, rants, docs).
4. RUN MY TASK AUDIT. Build the Level 2 table from what I told you, mark every row /goal, /loop or keep-human, and give me my top 3 automations ranked by minutes saved per week. Then fill out the conversion spec template from the playbook for my #1 task and build it with me.
5. PICK MY REVENUE LANE. Lane A (sell the output) or Lane B (growth engine), based on my situation. Design ONE engine with me, including the exact /goal prompt I will run, modeled on the enrichment engine in the playbook.
6. SHOW ME MY LEVEL 4 AND 5. One paragraph each: what my "build once, sell many" product would be, and which three boxes from the distribution playbook I should ship this week.

Rules: use the playbook's definitions, not generic advice. Every recommendation must name a concrete next action I can do in under an hour. If I am vague, push back. Do not move to the next step until I confirm the current one. Start with question 1 now.

Works in claude.ai with the PDF attached, or in Claude Code with this file in the folder. If you're in your business brain already, skip steps 1 and 3, it knows.

# The only mistake is staying there.

You know exactly which level you're on right now. Context, then workflows, then revenue, then scale, then distribution. Then the red arrow, and it all spins.

---

## Keep going:

- The Vault (free Telegram): every cheat sheet and prompt I publish, including updates to this one.
- AI Edge on YouTube: the video this guide comes from, plus the Slack agents video teased at Level 4.
- @milesdeutscher on X for the daily operator notes.