

Agentic Living

A Field Report from the Next Era

by Enzo Duit

“While we delay, life passes.” — Seneca, Letters to Lucilius

On the Night in the Mountains

There are moments when you understand something so completely that your body reacts before your mind has time to process it.

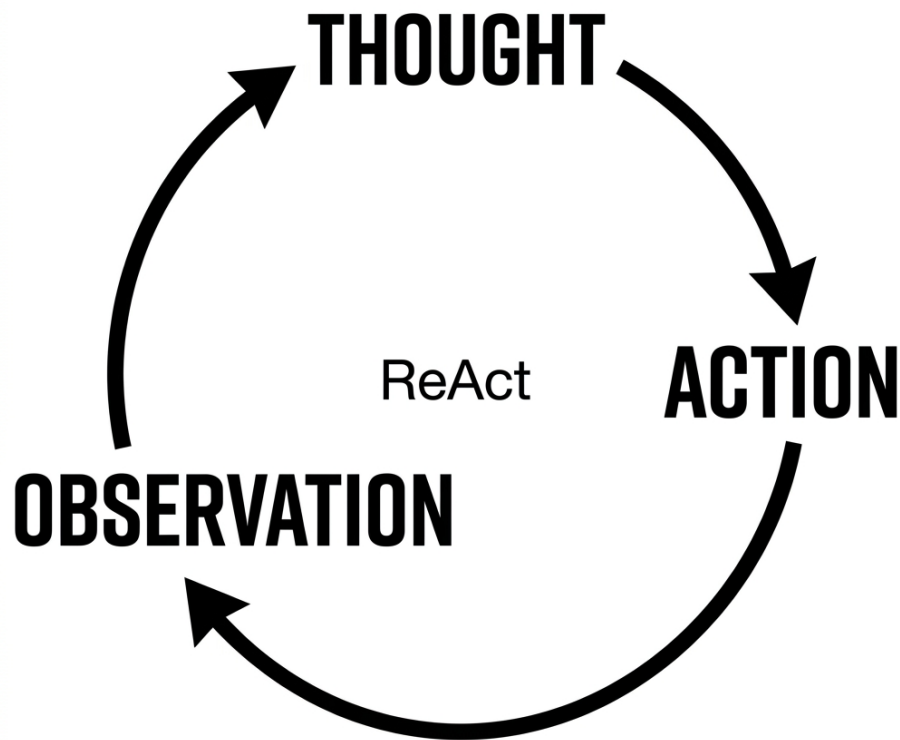
I remember one of those nights. Zell am See — one of my favourite places in the Austrian mountains — January 2025. I had been reading a research paper on my reMarkable. The ReAct framework. Reasoning and Action. A paper describing how an AI system could reason through a problem, take an action, observe what happened, and reason again — in a loop, autonomously, indefinitely.

I was completely shaking.

My heart rate was way beyond normal — I track it with my Garmin 24/7. The kind of heart rate you get before a race, or in a moment of real fear. I needed to go outside. Not for air — I needed to move, because the thoughts spinning up in my head were too fast for a sitting body.

It was a beautiful night with the moon shining bright. Freaking cold. Ten, fifteen degrees below zero. I just did not notice it, because I was internally on fire. I walked and talked to myself out loud, the way you do when you are trying to understand

something by hearing yourself say it. This paper described a system that could replicate whatever a human could do. Not a specific task. Not a narrow skill. The general loop: observe, reason, act. The loop that is the foundation of almost everything we call knowledge work.



The Agent Loop: Thought → Action → Observation. ReAct = Reasoning and Action.

If that loop could be automated, what exactly was left?

I had no answer that night. I walked until the cold finally reached me, then went inside. I did not sleep well.

What I did not know — what I could not have named in that moment — was that I had just crossed a threshold. I had not known it was there. I had not been trying to cross it. But somewhere on that walk, in the minus fifteen degrees, in the dark, reading a research paper by moonlight, I went from not

understanding what was now possible to understanding it completely. That night, without any vocabulary for it yet, I went from Level 0 to Level 1.

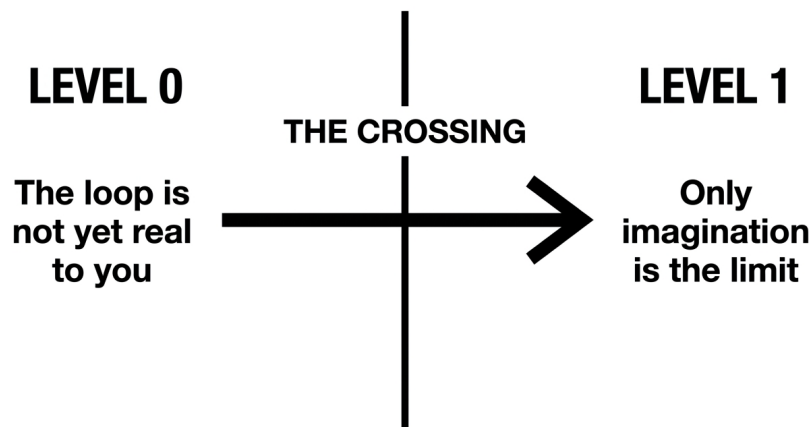
Two Levels

Let me explain what I mean by that. Because this is the frame through which everything else in this book should be read.

A chatbot responds when you speak to it. An agent runs while you sleep.

There are two levels of relationship with AI and agents. Most people are at Level 0. Almost nobody talks about this distinction. But it is the most important thing I know about the moment we are living through.

Level 0 is where you don't yet know — really know — what AI and agents can do. You may have seen demos. You may use ChatGPT. You may think you understand it reasonably well. But you have not yet felt in your body what it means for a system to build something complex, autonomously, from a single instruction — while you are doing something else entirely. You haven't watched it fail, correct itself, and produce a result that you would not have reached faster alone. Until you have experienced that, you cannot fully believe it. And because you can't fully believe it, you can't actually participate in what is being built around you. You are watching the game from outside the stadium.



The Crossing: Level 0 → Level 1

Level 1 is the moment you know. Not as an idea — as an experience. You’ve built something. You’ve watched something run while you weren’t watching. You’ve felt — in your body, not just your mind — that the only limit is what you can imagine. At that moment, everything you learned about how to code, how to execute technically, how to do the things that execution requires — it is worth almost nothing. Not because those skills are useless in isolation. But because everyone at Level 1 has access to the same capabilities. The barrier that used to separate the people who could build from the people who couldn’t — it’s gone. What’s left is not technical ability. What’s left is the quality of your thinking, the originality of your goals, and your capacity to turn imagination into something real.

Here is the part that surprises most people when I tell them: the market at Level 1 is harder than the one you left. Not easier. When everyone can build fast, the people who are ahead are not ahead because they can build — they’re ahead because they know what to build, and how to make it work commercially. At Level 1, you compete on judgment and monetization against a field where everyone is AI-enabled. It is a tougher game.

But here is what matters more than that: at Level 0, you are not playing at all.

Not because the game is unfair. Because you don't yet know it exists.

Why I Am Writing This

I crossed from Level 0 to Level 1 alone, in the mountains, reading a paper I'd found on my own.

What followed — the year I am about to describe — is what Level 1 actually looks like in practice. Not as a concept. As a way of running a business, running an ultra in Patagonia, fixing a server crash from a beach in Spain, having coffee in the Argentine highlands while three companies kept running without me at a keyboard.

But I also started teaching the crossing.

At IMC Krems in Austria. At Queensland University of Technology in Brisbane, Australia. Three-day workshops. People from different backgrounds, different levels of technical familiarity, different relationships with technology. The goal was always the same: by the end of the third day, everyone in the room would be at Level 1.

100% success rate across every cohort.

“If I can do it, everybody can — thanks to the technology we can access nowadays.” — Sofia, participant, QUT Brisbane workshop 2026

That number is the most important sentence in this introduction. Not because it proves I am a good teacher. Because it proves the crossing is learnable. It is not a talent. It is not reserved for developers or founders or people who grew up thinking about

systems. It is a threshold, and there is a path across it. I have walked people across it enough times now to know exactly where the difficult parts are, and what helps.

Peter Steinberger — the Austrian entrepreneur who created OpenClaw, the agent platform that made my year-long experiment possible — has a phrase I keep coming back to: “Only imagination is holding us back.” He said it in the context of what his software could do. But I think it describes Level 1 exactly. Not just the capability. The mental shift. The moment when you stop asking is this possible and start asking what should I build.

That is the shift this book is trying to produce.

My goal — with Agentic Living, with the workshops, with everything I have built around this — is to get as many people as possible to Level 1. Not because I think everyone will thrive there. The market is harder than before, and I won’t pretend otherwise. But because the alternative — staying at Level 0 in a world that is reorganising itself around Level 1 thinking — is to be excluded from a transformation that is already underway.

The Zell am See night was the moment I stopped being excluded. Everything that follows is what I found on the other side of that walk.

One note on who this is for. Some readers are building their own companies. Others are doing their best work inside existing ones. Both are at Level 0 right now. Both can cross. The destination looks different: the entrepreneur’s Level 1 is a product or company that runs largely without them in the execution. The employee’s Level 1 is a role that accomplishes more than any role before it — because everything that does not require their judgment is handled, and what remains is entirely their thinking. The crossing is the same. What you find on the other side is yours to define.

How This Book Is Structured

The framework you will encounter in this book — I call it AGENT — is not a productivity system. It is a map for the crossing from Level 0 to Level 1, and for what to do once you're there. The name is the acronym: **A**wareness, **G**rounding, **E**limination, **N**omination, **T**ranslation.

A — Awareness is the first move: understanding that the levels exist, that you are at one of them, and what the distance between them actually means for your work and your life. You cannot cross a threshold you cannot see.

G — Grounding is building the infrastructure that makes Level 1 real for you — not theoretical, not something you've watched someone else do, but running on your behalf, in your context, producing results you own. This is not a technical exercise. It is a trust exercise.

E — Elimination is stripping away everything that does not require your judgment. Not permanently — just starting with the minimum viable thing that proves the concept works. Find the core. Strip the rest. That is where Level 1 starts.

N — Nomination is deputising responsibility: not individual tasks, but roles. The parts of your working life that could be held permanently by an agent — because the context is clear, the standards are defined, and the output can be evaluated. Nomination is how Level 1 scales.

T — Translation is what you do with the hours that nomination returns. Not productivity — presence. Creating the conditions where your best thinking actually happens, and then directing that thinking back into the system.

The chapters that follow are organised around those five steps. Every story in this book is an example of what they look like in practice — the successes, the failures, the moments where the system corrected itself at 3am while I slept, and the moments where nothing worked and I had to figure out why.

Read it as a field report. Not as a manifesto. Not as a how-to. As notes from someone who crossed the threshold and kept walking.

What I found is worth the walk.

A	AWARENESS: See the loop clearly
G	GROUNDING: Build the infrastructure
E	ELIMINATION: Strip what doesn't need you
N	NOMINATION: Deputise responsibility
T	TRANSLATION: Reinvest the freed hours

The AGENT Framework: A·G·E·N·T

Chapter 1: On What Becomes Possible

Let me start with the thing that will be hardest for you to believe.

Early 2026. I had a brand new website — blank slate, no visitors, no reputation, no content, no history on the internet whatsoever. I gave an agent a single objective: get this website to the top of AI-powered search results for a specific topic.

I did not write a single word myself. I did not make a single decision about what the content should say. I did not change a single setting, pick a single keyword, or review a single draft.

I gave the agent the goal, described what a good result would look like, and let it work.

Six days later, the website ranked first on Perplexity — one of the leading AI-powered search engines, where millions of people now go to find answers — for the target topic.

While this was happening, I was in Patagonia. Running.

I want to be precise about what happened, because the precision is what makes it credible.

The agent chose what to write about. It analysed what questions were actually being asked in that topic area, identified where the gaps were, decided which angles would produce results. It wrote the content — not templated filler, but structured, specific pages that answered real questions. It handled the technical requirements: the formatting, the structure, the signals that AI search systems use to evaluate quality. It published everything. It

monitored what was working and kept going. No brief for each piece. No approval step. No human in the loop between the goal and the outcome.

One clarification that matters here, before we go further. When I say the agent chose what to write — that is true. But it did not choose freely.

I set the topic. I defined the context: what information the agent had access to, what it could draw on as the factual basis for everything it produced. That context was real — not generated, not interpolated, not approximated. Actual knowledge, actual expertise I had accumulated about the subject. And within that context, one constraint was absolute: use only what is real. No invented claims. No manufactured statistics. No opinions I had not actually formed, presented as if they were mine.

The agent had autonomy inside a defined space. The space itself — the goal, the context, the constraints — was entirely my judgment.

This is not a minor technical footnote. It is the difference between delegation and abdication. At Level 1, you are not releasing the steering wheel to something that drives wherever it wants. You are defining the road. What travels along that road, and how fast, and with what precision — that is where the agent operates. What the road is, where it goes, and what the guardrails are: that remains yours.

This is how it has to work — not just ethically, but because the output has your name on it. The constraint is what makes the output yours. The constraint is what allows you to stand behind it.

This is not a story about a clever tool. It is a story about what happens when “delegating a task” and “doing it yourself” stop being meaningfully different things — when a system can hold a

goal inside the space you have defined, make the decisions required to pursue it, and produce results that would have taken a team of specialists weeks to produce by conventional means.

If you are at Level 0 reading this, that description probably triggered one of two reactions. Either disbelief — that can't really work like that — or a quieter, less comfortable feeling: the sense of watching something move very fast while you are standing still. Both reactions are legitimate. Neither one changes what happened.

Here is what I have noticed when I describe this to people who have not yet experienced Level 1. There is always some version of the same response. It sounds like this: "Yes, but that's a specific kind of task. It doesn't apply to what I do."

I have heard this from lawyers, doctors, founders, artists, teachers, consultants, NGO directors, architects, journalists, and university professors. Every field has its version of the same objection. The specific task changes. The structure of the reasoning stays identical.

What the objection is actually saying — if you listen beneath the words — is this: I can imagine AI doing that particular thing, but I cannot imagine it doing the thing that matters to me.

That is a completely rational response. If you have not seen it work in your domain, on your kind of problem, producing your kind of output, you have no basis for believing it can. The demo videos do not count. Someone else's story does not count. The only thing that counts is watching it do something for you — something real, in a context you care about, producing a result you would have had to produce yourself.

The purpose of this book is to give you that experience.

Not by explaining it more convincingly. By building it, one step at a time, in your own context.

But before we get there — and I am going to ask you to do something at the end of this chapter that will take fifteen minutes and will matter more than you expect — I want to make the real claim clearly. Because the story about the website is not the point. The website is just a container. The point is what the story reveals about what has changed.

What changed is not that AI can write content. That has been true for years and mostly it produces content that nobody should read. What changed is the layer above the content — the layer of judgment. Deciding what to write. Deciding who it is for. Deciding what matters and what does not. Deciding when something is good enough and when it needs another pass.

That layer — the judgment layer — was, until recently, the part that required a human. The execution sat below it and required tools and time. Humans supplied the judgment; the tools supplied the throughput.

What is happening now, and what the Perplexity story illustrates, is that the judgment layer is also being automated. Not perfectly. Not in every domain. Not without careful goal-setting from a human. But increasingly, and in a growing number of contexts, a system can be given a direction and find its own way there — including making the choices along the way that used to require human intelligence.

That is the structural shift. Everything else — the lifestyle stories, the experiments, the tools and workflows and frameworks — is downstream of that one fact.

One measure captures the pace of this shift. METR — an independent AI research organisation — tracks what they call the “time horizon”: how long an agent can work reliably on a complex task before needing human input.

Year	Reliable autonomous operation
2019	~1 minute

Year	Reliable autonomous operation
2022	~8 minutes
2025	~1 hour
2026	~4-6 hours
Soon	Days / weeks

Source: METR Time Horizon 1.1, January 2026. Capability doubling approximately every 7 months.

An agent that works reliably for six hours is qualitatively different from one that needed checking every eight minutes. What required constant supervision becomes something you can hand off completely. And what you can hand off completely is not where your attention should go.

And here is the implication that most people resist when they first encounter it:

If the judgment layer can increasingly be held by something else, what is left for you?

Not which jobs will be automated. Not which tasks will disappear. What is the right kind of work for a human being — given that the execution, and much of the decision-making within the execution, can now be handled?

I spent the last year and a half running an experiment to answer that question. Not in theory. In practice, across four countries, two ultramarathons, a server crash fixed from a beach, and a year's worth of professional work done from a phone.

The short answer I arrived at is this: setting the goal and taking responsibility for the outcome. Those two things — the beginning and the end of any real work — remain irreducibly human. Everything between them is, in principle, not where your attention should go.

For most people in knowledge work, “everything between those two things” describes the majority of their working day.

What gets returned to you, when that majority is handled by something else, is not just hours. It is a different quality of presence with the work that actually matters — the capacity to think more clearly, to go further on the problems that deserve depth, and if you choose it, a different kind of life running underneath.

That is what Level 1 actually looks like. Not a shortcut. A structural shift in what your attention is for.

Your First Step

Before you read the next chapter, stop here.

Open a conversation with any AI — Claude, ChatGPT, Gemini, whatever you have access to. It does not matter which one. Type this exact prompt:

“I want to write a press release dated 2035. It announces something I built or created — a product, a service, a system, a change — that transformed how [your field / your work / your life] operates. Don’t write it yet. Interview me first. Ask me one question at a time until you have enough to write something real.”

Then answer honestly. Do not filter yourself. Do not evaluate whether what you’re describing is realistic. Do not try to sound impressive. Just answer the questions as if you are describing something that already exists.

You do not need to know how to build what you describe. The point of this exercise is not the press release. The point is the questions the AI will ask you — because they will be questions you have not been asked before, and your answers will surprise you.

When you are done, read what it produced. Not to critique it. Just to feel it.

Something in you will respond. Pay attention to that.

Keep the press release. We will come back to it.

SCAN → The challenge — 2035 Press Release Method agentic-living.com/press-release	 SCAN
 SCAN	← SCAN An example — SleepGuard: A Vision from 2035 agentic-living.com/press-release-example

→ If you are working within a company or organisation rather than building your own product: your press release does not need to be about a startup. Imagine it is 2035 and you are in your current role — or the version of it you would be proud to describe. Write the announcement of what you accomplished. What changed in how your work gets done? What did you deliver that was not possible before? What became possible in your position that nobody had managed before you? Let the AI interview you the same way. Your answers will be just as surprising.

Chapter 2: On Why Most People Never Cross

The question I get asked most often is not “what is AI capable of?”

Most people have a rough answer to that. They have seen the demos. They have read the articles. They know, abstractly, that something significant is happening.

The question I get asked most often is: “Okay, but what do I actually do?”

It is a more honest question than it sounds. What it is really asking is: I believe this is real, but I do not know where to start, and I am not sure I am the kind of person who can do this. The first part is an information problem. The second part is something else — and it is the part that keeps most people at Level 0 indefinitely.

I want to spend this chapter on that second part. Not the information. The invisible barrier underneath the information.

There are three reasons most people never cross from Level 0 to Level 1. They look like practical problems. They are not.

The first reason is that they are waiting to understand it before they try it.

This sounds reasonable. In most areas of life, understanding something before doing it is the right order. You study medicine before you practise it. You learn the language before you travel. The understanding comes first; the doing follows.

With AI agents, this order is reversed. You cannot understand what they are capable of by reading about them. You cannot understand them by watching someone else’s demo. You can only

understand them by doing something with them — and the doing will, almost certainly, feel uncomfortable and imprecise and not quite right the first time.

The reason the 3-day workshop works with a 100% success rate is not that three days is enough time to explain everything. It is that three days is enough time to force people to do something — and the doing produces an understanding that no explanation can replicate. By the end of Day 1, most participants are not at Level 1 yet. But they have done something. And the doing changes the question from is this real? to how far does this go? That shift is the beginning of the crossing.

If you are waiting to understand it first, you are waiting for something that will not arrive on its own.

The second reason is fear — that something could go terribly wrong.

This fear is often unnamed. But it shapes almost every early conversation about agents. You have probably read the stories: AI systems accessing personal data without permission, taking actions nobody asked for. There was significant press around specific platforms — concerns about AI going wild, doing unexpected things with private data. Some of those stories are exaggerated. Some are not. The concern is legitimate.

The answer is not to dismiss the fear. It is to design around it by starting somewhere where the cost of going wrong is effectively zero.

The first thing I automated was not a business problem. Not anything company-sensitive. It was my Garmin data — six years of running activities: GPS tracks, heart rate records, pace data, elevation profiles. Something entirely mine. If the agent misread a pace number or produced a wrong training summary, the

consequence was a wrong number in a report I would review anyway. No irreversible action possible. No sensitive data. Nothing that belonged to anyone else.

But from that starting point, something happened I had not fully anticipated. The agent took the raw running data and built something real: a short documentary of my training season, a coaching summary, a live page on a public domain. I had built software end-to-end — from raw data all the way to deployment, publicly visible — without being a developer. It was mine to own entirely. No company involvement, no permissions required, no risk to anyone else. And because the stakes were low, I could see exactly what the agent did and where the limits were.

That experience changed something. Not just because it worked — but because I could hold full responsibility for the outcome.

Start where failure is free. Start with something personal, something yours entirely, something where a wrong output teaches you more than it costs you. The Garmin experiment is the template.

The third reason is the most subtle, and the hardest to fix.

Most people, when they try to give an agent a task, discover that they cannot describe it clearly enough for the agent to execute it well.

This feels like a problem with the agent. It is not. It is a problem with the clarity of their own thinking.

When you have been doing something for years, you develop unconscious competence — the ability to do a task well without being able to explain exactly how you do it. The judgment is so integrated that it no longer feels like judgment. It feels like instinct. And when you try to hand that task to an agent, you discover that “do it the way I would do it” is not a specification. It is a hope.

The process of writing clear instructions for an agent — good constraints, real context, explicit criteria for what success looks like — is also the process of articulating your own expertise to yourself. It is uncomfortable and slow the first time. It is one of the most valuable things you will do.

Because once you have articulated it — once the implicit has become explicit — two things happen. The agent can execute it. And you understand your own work more clearly than you did before.

This is one of the paradoxes of Level 1 that nobody prepares you for: the technology forces a precision of thinking that most people have been avoiding for years.

I want to name one more thing that sits beneath all three of these, because pretending it does not exist would make this book less useful.

Some part of most people is afraid of this.

Not afraid of the technology failing. Afraid of the technology working. Afraid of what it means for how they have been spending their time, for what they thought their expertise was worth, for the story they have been telling themselves about why their work requires them specifically.

This fear is legitimate. It is pointing at something real. The structural shift I described in the first chapter — the judgment layer moving up, the execution layer moving to agents — does change what expertise means. It does change what your time is for. It does change the story.

The question is not whether the story changes. The question is whether you get to be the one who changes it deliberately, or whether it gets changed around you while you were waiting to understand it first.

Your Second Step

Take out the press release you wrote at the end of the last chapter.

Read it once.

Now — before you open any AI — close the screen. Think offline, without inputs. What is the one thing that could actually go terribly wrong with this vision? Give yourself a few minutes. Let your own thinking happen first, before the AI starts shaping the conversation. When you have a clear sense of the main risk, then open the AI.

Now open a new AI conversation and type this:

“I’m going to share something I wrote about a future I want to build. I want you to play devil’s advocate. Assume it is the year this launched — and something went wrong. Not everything fell apart, but one thing did. It could be a mistake I made. It could be something I did not anticipate. It could be a way someone else was harmed. Ask me questions until you understand my vision, then tell me the one most likely failure. Be specific and honest — I can handle it.”

Share your press release with the AI. Let it read it. Let it ask questions. Then let it tell you the most likely way your vision fails.

This is the Black Mirror session from the workshop — compressed into fifteen minutes.

The purpose is not to talk yourself out of what you wrote. The purpose is to make what you wrote more honest. A vision that has survived its own stress test is worth pursuing. A vision that has not is just optimism.

After the AI tells you the most likely failure, write one paragraph — in the same document as your press release — describing what you would do to prevent it.

You now have a vision and a first principle for how to build it responsibly. That is not a small thing.

SCAN → The challenge — The Black Mirror Session agentic-living.com/black-mirror	 SCAN
 SCAN	← SCAN An example — SleepGuard in Crisis: How It Could Go Wrong agentic-living.com/black-mirror-example

→ If you are working within a job: your press release was about your role, not a startup. Run the same exercise: what is the most likely way this vision fails? Maybe it is that you build something your organisation does not allow you to use. Maybe it is that you automate the wrong thing and lose the confidence of your team. Find that specific failure. Then write one paragraph on what you would do to prevent it.

Chapter 3: On Grounding — Building the Infrastructure of Trust

Before you build anything, you need to trust the thing you are building with.

That sounds obvious. But in practice, it is the step most people skip — and the reason most early experiments with AI agents end badly. Not because the technology failed. Because the person setting it up had not thought carefully enough about what they were actually connecting, to what, with what permissions, and why.

This chapter is about building that trust correctly. Not by hoping for the best. By designing the constraints first.

The Fear That Is Not Irrational

When I describe what I have built — an agent that runs continuously on a server, with access to my files, my emails, my calendar, my code, my communications — the reaction I get most often is some version of: “Doesn’t that mean it can do anything? Doesn’t that mean you’ve lost control?”

This is the fear that keeps more people at Level 0 than any other. And I want to be direct: it is not an irrational fear. It is pointing at a real risk. The question is not whether the risk exists — it does — but whether you can design around it intelligently.

The short answer is: yes. But only if you are deliberate about it before you start.

Here is what most people imagine when they think about an AI agent with access to their system: a single large entity with unrestricted access to everything, making decisions they cannot predict, reading what it should not read, changing what it should not change, acting without asking. Something like handing your house keys to a stranger and saying “please tidy up” without specifying what tidy means, what you want touched, and what is absolutely off limits.

That image is wrong. But it is only wrong if you make it wrong — if you design the system so that it is not that.

The Isolation Principle

The first thing to understand is that a well-designed agentic system is not one entity with unlimited access to everything. It is a set of specifically scoped capabilities, each with defined boundaries, each operating only within the space you have intentionally given it.

Think of it not as handing over the house keys, but as giving a specific person a specific key to a specific room, with specific instructions about what they are allowed to do in that room.

The agent I run on my server does not have unrestricted access to my computer. It has access to a defined environment: a server that I control, with connections to specific services that I have deliberately chosen to connect. When a new connection is needed — a new service, a new data source, a new capability — I add it intentionally. Nothing connects without a decision.

This isolation is not a limitation on what agents can do. It is what makes them safe to run. And critically, it is what makes the output trustworthy — because you know exactly what the agent had access to when it produced it.

The Read-Only First Principle

The second thing to design correctly is permissions. Specifically: start with read-only access, and earn write access over time.

This is the practical equivalent of what you would do with any new employee. You would not give a person you hired last week the ability to delete your company's database. You would give them the ability to read it, understand it, work with it — and only expand their permissions as trust was built and demonstrated.

The same logic applies here, and it is surprisingly easy to implement. Most services and APIs offer both read and write access as separate permission levels. When you first connect an agent to a service — your email, your calendar, your files, your business data — connect it read-only. See what it does with that access. Evaluate the outputs. Build confidence.

Only when you have seen the agent handle read access well — when you understand how it interprets the data, what it extracts, what it flags, what it ignores — should you consider giving it write access. And when you do, start with low-stakes write actions: drafting a message that you approve before it sends, creating a document in a sandbox folder, logging a summary to a file you review daily.

Here is a concrete example of what this looks like in practice. My email connection is configured to create drafts — not to send them. Even if the agent composes a perfectly written email, even if the context makes the response obvious, even if everything about the situation calls for it to send: it technically cannot. The send action does not exist in the permissions I granted it. The last click is always mine.

This is not a trust issue with the agent's writing. It is a design decision about which actions should require a deliberate human decision. Sending an email is irreversible. Once it leaves, it leaves.

That is exactly the kind of action that should never happen without a human at the final step — regardless of how much you trust the system or how good the draft is.

Technical restrictions are more reliable than behavioral ones. Do not rely on the agent “knowing” not to cross a line. Make it technically impossible for it to cross that line. The best-designed systems make the right behaviour the only possible behaviour.

The instinct to “give it everything so it can really help” is the instinct that gets people into trouble. More access does not produce better results. Appropriate access, well-defined, produces results you can trust.

The Data Perimeter

The third design decision is what data the agent has access to at all.

Not every piece of data you own should be available to an agent. Sensitive personal data, confidential client information, financial records, legal documents — these have higher stakes if mishandled. They require higher trust before you connect them — or you just work with IDs. In many cases you do not need the actual sensitive data; you only need a reference to it. An order number, a client reference, a file path. The agent can work with the identifier without ever seeing what it points to. Same functionality, smaller security surface.

The practical recommendation is to start your first experiments with data that has low sensitivity and high volume. Email newsletters, public-facing content, historical reports, meeting notes from internal planning sessions — these give the agent enough to work with to produce genuinely useful outputs, without exposing anything you could not recover from if something went wrong.

As the system proves itself — as the outputs are consistently correct, consistently within the space you defined, consistently aligned with what you would have done yourself — you expand the data perimeter. Deliberately. Not because the agent has earned it in some abstract sense. Because you have evidence that it handles data the way you want it handled.

The Flow: What Goes In and What Goes Out

There is a practical mental model that makes the design of a secure agentic system much clearer. Think about two categories: what flows into the agent automatically, and where the agent can communicate something out.

Incoming email is the clearest example of automatic input. If you connect your inbox to an agent, any person in the world who sends you an email is contributing information to your agent's context. You have no control over who sends you email — which means you have no control over what information arrives. That is fine. But it means you need to think carefully about what that information can influence or trigger.

The security question shifts when you apply this lens: not just “what does the agent have access to?” but “what can get in, and what can get out, and how?”

The answer to what can get out is the more important one. Every channel through which the agent can communicate externally — email responses, published content, API calls, messages sent on your behalf — is a potential point of failure. If the agent can send email, a badly framed instruction could result in a message you did not intend to send. If it can publish content, a misconfigured constraint could result in content you did not intend to publish.

The practical answer: restrict the output channels before you worry too much about the input channels. Be deliberate about what the agent can do in the world — what it can send, where it can write, what it can publish, who it can contact. Expand that list incrementally as trust is built.

Think of it as a semi-permeable membrane: broad input, narrow and controlled output. The world can tell the agent many things. The agent speaks back to the world only through channels you have explicitly opened.

AI as Your Own Security Officer

I want to tell you something that happened on July 17, 2026. Not because it is a horror story — it is not — but because it illustrates what happens when you get the design right.

My agent ran its daily infrastructure check. It found something. The portal where the agent itself operates — where its task queue sits, where its monitoring data lives, where its decisions are logged — had no access control on it. Anyone who knew the address could open it.

This was not a catastrophic breach. But it was a real gap — and left uncorrected, it would have been a real risk. The agent flagged it, protected the portal, extended its monitoring to seven domains it had not been watching before, and built a permanent alert layer so that the same thing could not happen again.

By the time I received the summary message, the portal was already protected.

I have had this happen several times. The agent finds a security issue before anyone else does, fixes it, and tells me what it fixed. The summary arrives after the problem is solved.

This matters for two reasons. First: it shows that even with a lot of experience, things can still slip through. A portal left exposed is exactly the kind of thing that gets missed in the noise of running multiple systems at once. Second: it shows that AI can find the things that go wrong faster than a human can — if it is given access to the right data and a clearly defined responsibility.

The key word there is responsibility. I did not give the agent a task: “check for this one vulnerability.” I gave it a role: Data Security Officer. The distinction is everything. A task ends. A role continues. A role looks for problems you did not think to specify.

This is something you can implement for your own setup — early, before things get complex. An agent with read access to your infrastructure and a clear mandate to monitor and flag can do what a human security officer does, but continuously, without forgetting, without needing to be reminded. And unlike a human, it can act the moment it finds something — not in the next meeting, not after a ticket is raised.

The infrastructure you build is not just the thing the agent uses to do your work. It is also the thing the agent uses to keep your work safe.

What This Looks Like in Practice

Let me make this concrete with how I actually started.

The first thing I connected was my running data. Ten years of activities from Garmin — GPS tracks, heart rate records, pace data, elevation profiles. The stakes of something going wrong: effectively zero. If the agent misread a pace number or drew an incorrect conclusion about my training load, the consequence was a wrong number in a report I would review anyway. No sensitive information. No irreversible action possible. High volume of data that could teach the agent a lot about how to work with it.

From there, I expanded: calendar data (read-only), email (read-only, for analysis and drafting), then specific file folders, then specific APIs. Each new connection was a deliberate choice. Each one came after I had seen the previous connections work correctly.

The first time I gave the agent write access — the ability to actually publish something, not just draft it — was months into the experiment, for a specific use case I had watched it handle perfectly in read-only mode for weeks. And even then, the first published outputs were things I reviewed before they went live.

This is not how technology demos work. Technology demos show you the full capability immediately. This is how responsible deployment works: trust earned incrementally, access expanded in proportion to demonstrated reliability.

The Fear, Answered

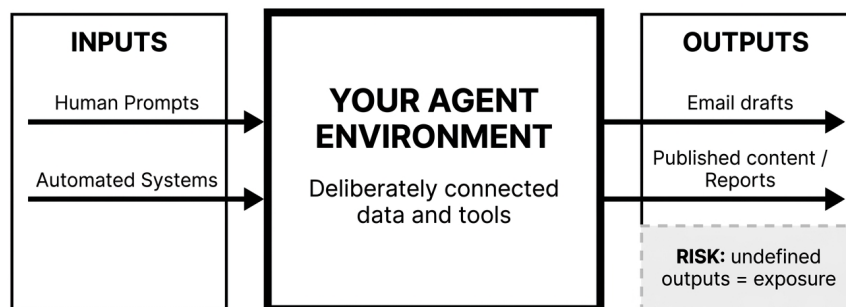
So: does connecting an AI agent to your systems mean it can go wild with your data?

No — if you design it correctly.

The isolation principle means it only has access to what you have explicitly given it. The read-only first principle means it cannot change anything until you have decided it is ready to. The data perimeter principle means you are not exposing sensitive information until you have built the evidence that it handles less sensitive information correctly.

What you are building, when you do this well, is not a system that does whatever it wants with your life. You are building a system that does exactly what you have defined, within exactly the space you have defined, and reports back to you so that you can see what it is doing. The autonomy is real. The constraint is also real.

That combination — genuine autonomy within genuine constraints — is what makes it useful and what makes it safe. One without the other is either useless or dangerous.



Your Agent Environment: inputs, outputs, guardrails

This is also why — if you have heard the OpenClaw stories and security concerns about AI accessing your computer — the right response is not to avoid it. It is to install it correctly. Not on your personal laptop or your work machine, where everything on the device is potentially reachable. On an isolated machine: a separate server or clean environment where the only access that exists is what you have deliberately granted. Nothing else connects without a decision. That isolation is not a workaround. It is the design.

The Cost Layer

There is one more category of risk that most people encounter the hard way: financial exposure. Two scenarios are worth understanding before you build anything.

The API key story. When you build an AI-powered website — anything that calls an AI model from a page a browser loads — you need an API key. Think of it as a password that tells the AI

provider: this request is from me, charge my account. Some developers, often newer ones, put this key directly in the frontend code of the website — the HTML and JavaScript that the browser downloads when someone visits the page. That code is visible to anyone who knows where to look. Thousands of people do this. And thousands of others actively search for exposed keys. When they find one, they use it at the original owner's expense. People have woken up to bills of hundreds or thousands of dollars from usage they never authorised. The fix is simple: API keys never go in frontend code. They live on a server, never sent to the user's browser. But the story is worth knowing because it illustrates the core principle: the things you choose to expose are the things that create risk.

The endless loop. The ReAct loop — Thought → Action → Observation, repeating until the goal is reached — means the agent can, in theory, run indefinitely. If something goes wrong — an unclear goal, a task that keeps generating subtasks, a situation the agent cannot resolve — it keeps running. And you pay for every step. Input tokens. Output tokens. Every intermediate reasoning step the model generates. A runaway agent can accumulate costs quickly, without you knowing it is happening.

You pay for what the model processes. Every word in, every word out, every reasoning step in between — these are billed in units called tokens. Roughly one token per syllable. A short conversation costs cents. A runaway loop that runs for hours can cost significantly more.

The guardrails. Build your financial protection before you need it — the same way you build the security perimeter before you connect anything sensitive.

At the payment level: Use a card with a hard spending limit — a dedicated card, a virtual card, or a prepaid card with a ceiling you set. Not your main card with no cap. When the card stops,

something may break. That is exactly what you want: a hard stop that tells you something went wrong, rather than an unbounded bill arriving later.

At the provider level: Every major AI provider offers spending limits, usage alerts, and per-period caps in their account settings. Set them before your first experiment. A notification at €10 spent. A hard limit at €50. These are standard features — not using them is the equivalent of running a server with no monitoring.

Multiple layers — card limits, provider limits, usage alerts — mean that any single failure has a ceiling. You decide the maximum scale of the bad outcome before the first line runs. The people who ran into the horror stories almost all had the same thing in common: they had set no limits at all.

Your Third Step

You have your 2035 vision. You have stress-tested it against its most likely failure. Now you design your first real experiment.

The constraint: it must be read-only. The agent can observe, analyse, summarise, extract, and report — but it cannot change, post, send, or publish anything.

Open an AI conversation and type this:

“I want to design my first real AI experiment. It needs to be read-only — the AI can look at things, summarise things, and tell me things, but it cannot change or send anything. Here is what I have access to that I would be comfortable with an AI reading: [list 2-3 things — your calendar, your email, your files, your notes, a dataset you work with, whatever applies]. Given that, what is the most

useful thing an AI could do for me right now that would show me something about my own work that I currently cannot see? Design the experiment.”

Let it design the experiment. Then ask it one follow-up question: “What would I need to give you access to in order for this to work, and how would I do that safely?”

You do not have to run the experiment yet. But by the end of this conversation, you will know what your first real experiment is, what access it requires, and what the output will look like.

That is grounding. You are no longer imagining Level 1. You are designing the path to it.

→ If you are within an organisation: before you list what you would be comfortable with an AI reading, add one check: is any of this controlled by your employer? Company email on company servers, shared drives, enterprise tools — check with IT or your data security team before connecting those. The principles in this chapter apply fully. The perimeter is just more clearly defined from the start. Begin with what is clearly yours: your personal notes, your own documents, your calendar. The crossing starts there.

Chapter 4: On Elimination — The Minimum Viable Proof

You have a vision of what Level 1 looks like for you. You have stress-tested it against its most likely failure. You have designed an experiment that is read-only, bounded, and safe to run.

Now comes the part where most people stop again.

Not because they run out of courage. Because they run out of clarity. They sit down to build their first thing and immediately confront a version of the same problem: where exactly do I start?

The answer is elimination. Cut everything that is not necessary to prove the concept. Not permanently — just for now. Because the minimum viable thing is not a compromise. It is the fastest path to knowing whether the larger thing is worth building at all.

Shell vs. Brain — What to Actually Build

When you wrote your 2035 press release, you imagined something complete. That is exactly what the exercise was designed to produce. The trap is trying to build toward that destination all at once — and more specifically, building the wrong part of it first.

Most people, when they sit down to build, reach for what they can see. The interface. The name. The landing page that explains what the product does. The mockup that shows what it looks like. In the workshop, we call this building the Shell.

The Shell is not the product. The Shell is the container for the product. Containers that contain nothing are just boxes.

What you need to build is the Brain — the intelligence, the logic, the actual decision-making that is the value you promised. Strip away everything physical from your press release. Strip away the

name, the branding, the interface. What is left? What is the actual decision that needs to happen? What does the system need to know — and what does it do with what it knows?

The Brain is almost always the same thing: **data plus decisions**. What information goes in, and what output comes out?

The workshop example that makes this concrete is SleepGuard. The 2035 vision: a Neural-Weave Pillow that detects your mood and plays the exact audio frequency to induce sleep. The Shell is the pillow — the hardware, the neural interface, the physical device. Obviously impossible to build in 48 hours.

But the Brain? The Brain is: take signals about someone's current state, determine the right audio response, deliver it. Data — the user's state — plus a decision — the right frequency. Both buildable today, without any hardware, using an AI agent that asks a few questions and queues the right audio. Teams in the workshop call this the Sleep Concierge Agent. It does exactly what the pillow promised. It just does not have a pillow around it yet.

The Shell was 2035. The Brain was already buildable in 2025.

The same gap exists in almost every ambitious vision. Find the Brain in your press release. That is what you build first. Everything else can wait.

And never forget to challenge based on the principle you just learned.

What Happens When You Run It

The first time you run a real experiment — not a demo, not someone else's use case, but your task, your data, your output — something shifts.

It is not always a perfect result. Usually it is not. The first output will be slightly off in a way that teaches you something: either the task was not scoped narrowly enough, or the context you provided was not precise enough, or the output format was not what you actually needed.

Each of those is valuable. Because the correction required to fix the output is also the process of articulating more clearly what you actually want. And when you articulate that clearly enough for an agent to do it, you have done something that has a value beyond the agent: you have made explicit something you previously held only as instinct.

Run the corrected version. Evaluate it. Adjust again if needed. By the third iteration, most experiments produce outputs that are genuinely useful — not because the agent got smarter, but because your instructions got clearer.

This process — run, evaluate, refine, run again — is the texture of Level 1 in its early phase. It is not passive. It is not “let the AI do everything.” It is active collaboration, where your contribution is the clarity of your thinking and the quality of your goal-setting, and the agent’s contribution is the execution that makes your thinking visible.

The data from the experiment tells the same story:

Period	Interactive sessions	Autonomous (background)
March 2026	~73%	~27%
July 2026	~4%	~96%

In March, a message arrived every 3.3 minutes on average — constant direction required. By July, 96% of sessions ran autonomously, with no input at all. The curve is not the agent getting smarter. It is trust accumulating — and every nominated responsibility reducing the next interaction required.

Pro Tip: Quality Gates Before the Output Reaches You

There is one additional design decision that significantly improves the quality of what you get back — and reduces the back-and-forth that makes early experiments feel exhausting.

Define the quality criteria upfront, and tell the agent to check its own output against them before it sends you anything.

Without this, the default pattern is: agent produces something, sends it to you, you give feedback, agent revises, you evaluate again. This works — but every iteration requires your time and attention. You are the quality gate.

With criteria built in, the pattern changes. The agent produces something, checks it internally against the criteria you defined, and if it does not meet them, it revises before you ever see it. What arrives is already past the first round of review. You are no longer the first reader — you are the final one.

Here is what quality criteria look like in practice. For a daily email summary: “Check that every item is a direct quote or accurate paraphrase from an actual email. Check that the three most important items are listed first. Check that the total summary is under 200 words. If any of these is not met, revise before sending.”

That is a quality gate. The agent runs it, not you. The result: less back-and-forth, higher quality from the first version, and more of your attention available for the things that actually require a decision from you.

The Question of What to Eliminate from Your Life

There is a second kind of elimination that becomes available once the first experiment is working.

Up until this point, elimination has been about scoping the experiment down. Now it becomes about looking at your actual working day and asking: what here should not be mine?

Not what is boring. Not what is tedious. What genuinely does not require your judgment?

This is a different question, and it is harder to answer honestly. Most people have spent years treating all of their work as if it requires them equally — because no alternative existed. When the alternative appears, suddenly some of the work reveals itself as never having required a human at all. It was only being done by a human because there was nothing else to do it.

The candidates are usually obvious once you look for them: the recurring reports that follow the same structure every time, the research tasks that involve gathering and summarising information that already exists, the monitoring tasks that require checking something and flagging a deviation, the formatting and filing that happens after the real work is done.

These are not small things. For most people doing knowledge work, they account for two to four hours of every working day. That is one full working day per week — returned to you, if you are willing to design the system that handles it.

Eliminate from your own plate what does not belong there. This is not laziness. This is precision. The goal is not to work less. The goal is to work on the things that actually require you.

Your Fourth Step

Take your read-only experiment design from the last chapter and run it.

Before you start, write one sentence describing what success looks like. Not “it works” — something specific. “The daily summary contains the three most important things I would have

noticed myself, without me reading everything.” “The meeting notes capture every action item with the right owner.” “The research briefing gives me enough to make a decision without reading the original sources.”

That sentence is your evaluation criterion. At the end of the week, you will know whether you met it.

If you are unsure how to set the experiment up technically, use this prompt:

“I want to run a one-week AI experiment. The task is [your task]. The data source is [your one source]. The output should be [your one output]. I want to keep this as simple as possible — the goal is to test whether the concept works, not to build a complete system. I am not a developer and I do not want to write code. Please suggest existing AI tools and services I can use, and explain each step in plain language without technical jargon. What is the simplest way to get this running today?”

Let the AI walk you through the setup. Ask follow-up questions until the path is clear.

Then run it.

Whatever happens in that first week — success, partial success, complete failure — you will have crossed a line that most people never cross. You will have moved from imagining Level 1 to testing it, in your context, on your terms.

That is not a small thing. That is the crossing.

SCAN →

The challenge — The Minimum Viable Proof

agentic-living.com/mvp



SCAN



SCAN

← SCAN

An example — SleepGuard: From Vision to Brain

agentic-living.com/sleepguard

→ If you are working within a job: your MVP is not a product. It is proof that one task in your role can be handled better. Look at your working week and find the task that takes the most time and requires the least judgment — the one that exists because someone has to do it, not because it requires you specifically. Design the experiment around that one task. If it works, you have demonstrated Level 1 in your context. That demonstration is also what makes the conversation with your organisation possible.

Chapter 5: On Nomination — Deputising What Does Not Require You

You have crossed from Level 0 to Level 1. You have proven one thing: a system can do the thing. You ran an experiment, and you demonstrated that autonomous operation is real for you, not just in principle.

The question arrives immediately: what next?

The answer is Nomination. Not more experiments on the same scale as the first. The practice of identifying what should be handed to an agent permanently — not as a task, but as a responsibility — and growing the stack one nomination at a time.

What the Data Actually Shows

Let me give you the numbers first, because numbers are harder to dismiss than philosophy.

For the first four months of my experiment, the ratio of autonomous background sessions to interactive ones stayed roughly flat. Somewhere between five and thirty percent of all activity was automated — scheduled checks, cron jobs, reports that ran whether or not I was at a keyboard. No clear trend.

Then July: fifty-seven percent autonomous.

In a single month, the agent ran 728 autonomous sessions alongside 542 interactive ones. For the entire first four months combined, there had been 33 autonomous sessions total.

The progression makes the dynamic visible:

Period	Autonomous	Interactive	% Autonomous
March 2026	17	46	27%
April – June 2026 (combined)	16	—	—
July 2026	728	542	57%

This is not a gradual curve. It is a step function. March through June: 33 autonomous sessions total, the system mostly in scaffolding mode. July: 728. The nominations made across four months started compounding all at once.

This did not happen because of a single decision. It happened because of dozens of nominations made over four months — each one adding a layer to the stack, each one expanding what the system held without me. The July numbers are not the result of a moment. They are the result of a practice.

Nomination is how the stack grows.

Task Executor vs. Responsibility Holder

There are two ways people work with AI agents. Most people default to the first. The most powerful practitioners use the second.

The first way: task executor. You brief it; the agent completes it; you review it; the cycle repeats. Tasks work. They produce real results. But tasks are not what produces a 57% autonomous figure.

The second way: responsibility holder. You define the scope; the agent holds it continuously; it acts without being asked; it finds problems you did not specify.

The difference in results is not incremental. It is categorical.

Most people use agents as task executors. The more powerful use is as responsibility holders.

Here is the distinction in practice. The weak version of nomination sounds like this: “Monitor our domain’s performance and tell me if it drops.” This produces a task executor — a system that checks once, reports, and waits for the next instruction.

The strong version sounds like this: “Your responsibility is that this domain ranks first on Perplexity for our target queries. The metric is position. You can check this yourself. If we are not ranked first, investigate why and propose specific changes. If we are gaining ground, identify what is working and recommend how to extend it. You hold this outcome. Report to me only when you have a specific recommendation or when the situation changes materially.”

The first gives you a reporter. The second gives you an employee.

The Data Security Officer

The clearest example from my own system: the Data Security Officer.

I know what a Data Security Officer does. I know their checklists, what they audit, what they document, what they flag. So rather than giving my agent a checklist to run once, I gave it a job: “You are the Data Security Officer for this system. That is your responsibility. Hold it. Do what that job requires.”

The agent built a compliance profile. It created automated monitoring. It flagged exposures. It documented everything. The output was not a task completed — it was a function held.

On July 17 — and you will recognise this case from the Grounding chapter — the Data Security Officer demonstrated the difference directly. Running its daily check without being asked, it found something I had not specified: the portal where the agent itself operates had no access control on it. Anyone who knew the address could open it.

The agent flagged it, protected the portal, extended monitoring to seven domains, and built a permanent alert layer. By the time I received the summary, the problem was already solved.

A task executor would have run its check and moved on. A responsibility holder found something outside its assigned scope, understood why it mattered, and acted.

An agent holding a responsibility is not the alternative to an expert. It is the alternative to the gap.

If you are already running OpenClaw and want a ready-to-use interface for managing the digital employees you nominate, I have built an open-source framework for exactly this: → **Digital**

Employee Framework



. Fork it, adapt it to your stack, and

use it as the management layer for the roles you are building.

The Knowledge Principle

Here is the one thing that governs everything in this chapter — and the one most people get wrong.

You need to understand the role you are nominating well enough to evaluate whether the agent is fulfilling it correctly.

Not to do it yourself. You are nominating because the execution should not be yours. But you need to be able to read the output and judge it. You need to ask the right questions, challenge a result that looks wrong, and refine the standard over time.

No knowledge plus AI is dangerous. The agent will produce something, and you will have no way to tell whether it is doing it well or poorly. You cannot catch the moment things go wrong. You cannot improve what you cannot evaluate.

Sufficient knowledge plus AI is extremely powerful. You set the brief. You evaluate the output. You refine the criteria. The agent executes at scale; you hold the standard.

I knew what a Data Security Officer does. That knowledge is what made it safe to nominate that role. Not deep expertise — I am not a security specialist — but sufficient understanding to know when the agent's work was correct and to ask a sharp question when it was not.

This is the threshold: not the ability to do the job yourself, but enough domain knowledge to rate and challenge the output. If you cannot evaluate whether the nomination is working, you have not delegated a responsibility. You have abandoned one. These are not the same thing.

This principle also means something positive: the practice of nomination forces you to articulate your expertise. To brief a nomination properly, you must describe the role precisely enough for someone else to hold it. That articulation is valuable in itself. Many people discover, when they try to brief their first nomination, that they understand their own work more clearly than they did before.

Your Fifth Step

Look at the ongoing responsibilities in your working life — not one-off tasks, but things that need continuous attention: monitoring something, maintaining something, holding a standard over time.

For each, ask: do I understand this well enough to evaluate someone else doing it?

Pick the one where your answer is clearest. Brief it as a role, not a task. Write the strong version: define the outcome, give the agent the metric it can use to evaluate its own performance, and be specific about what good looks like.

Then open an AI conversation and type this:

“I want to brief an AI agent to hold a role with ongoing responsibility, not just complete a task. The role is [your responsibility]. Here is what I understand about what the role requires: [describe it as you would to a new employee]. Based on this, what would the agent need to have access to, what would it monitor or produce on a regular basis, and how would you suggest I design the brief so I can evaluate whether it is doing it correctly? I am not a developer and do not want to write code. Plain language only.”

Pay attention to what the AI asks you to clarify. Every clarification request is a gap in your current brief — and a gap the agent would have fallen into if you had not caught it first.

→ If you are within an organisation: your first nomination is not a product or a new system. It is an ongoing responsibility in your current role — something that needs regular attention, that someone competent could hold with the right brief. Name that responsibility. Brief it the strong way. The conversation is exactly the same.

What you do with the hours that nomination returns — how you translate the freed attention into real output — is the chapter that follows this one.

Chapter 6: On Translation — Making the Freed Hours Real

Here is the hardest question in this entire book. Not how to build the system — by now you have the tools to do that. Not how to nominate the first responsibility — you have the framework.

The hardest question is: once the system is running and the hours start coming back to you, what do you actually do with them?

This is not a small question. Most people who arrive at Level 1 expecting freedom discover something unexpected: the time that was previously consumed by execution does not automatically fill with better work. It fills with whatever is nearby. More email. More meetings. More responding to inputs. The same kind of low-quality busyness they had delegated, now replicated by choice rather than necessity.

Translation is the practice that prevents this. It is the answer to the question of what to do with what the system returns — and it is the reason this framework has a T at the end.

Where Your Best Thinking Actually Happens

Let me ask you something before we go further.

Think of the last time you had a genuinely original idea. A real one — something you had not read somewhere, something that came from connecting things in a new way. Where were you?

Almost certainly not at your desk.

Over the course of the experiment, I noticed a pattern I could not ignore. My best thinking was arriving in specific places: on trails, in coffee shops before I had opened anything, on long runs at altitude, in the mountains waiting for weather windows. Almost never in front of a screen.

This is not a mystical claim. It is a practical observation about cognitive performance. The kind of thinking that produces real insight — making connections, reframing problems, seeing what is missing — does not happen efficiently under distraction. And screens are distraction delivery systems. Not because of social media or notifications (though those are real), but because the interface is designed to keep you responding to inputs rather than generating original thought.

The screen is where I record the output. It is not where the output comes from.

Understanding this changed everything about how I structure my days.

The Three Phases

The rhythm that emerged, after months of trial and error, has three distinct phases. None of them looks like a conventional working day.

Phase one: offline thinking. Two hours, no screen, no inputs. For me this is a reMarkable and a pen — I find that writing by hand, on a device with no notifications, creates exactly the right conditions. But this is personal. For someone else it might be a notebook, a walk with no phone, a coffee shop before anything is opened. The specific tool matters less than the principle: no inputs, no responses, no screen demanding attention. The raw thinking happens here — problems worked through, directions decided, what needs to happen clarified. Nothing is produced for anyone. This is the substrate that everything else runs on.

Phase two: deep co-creation. Two hours with the agent, building or deciding something specific. Intense, absorbed, total focus. The offline thinking feeds directly into this — I arrive knowing what I want to do, which makes the co-creation faster and sharper. These sessions have a hard limit. After roughly two hours, the quality of judgment declines — the briefs get vaguer, the calls get lazier. The discipline is to stop at that point and not push through.

Phase three: away from the phone entirely. For me this is movement — long runs, Zone 2 training, time at altitude. That is my version. For you it might be something completely different: a drive, a swim, cooking, a long walk, an afternoon with no agenda. The point is not the activity. The point is that you are genuinely elsewhere, not half-present. The agents run in the background. The outputs accumulate. You are not managing them.

I run ultras, which means I spend a lot of time training at low intensity — what coaches call Zone 2. Heart rate controlled, legs moving, mind free. I stopped seeing this as time away from work a long time ago. It became the best thinking time available to me, because the systems were in place and acting on what I thought required only a few minutes, not a full session. But the principle — not the sport — is what matters here. Find the activity that gets you genuinely off the screen. The thinking will follow.

This creates a working rhythm that looks strange from the outside: four hours of high-intensity cognitive work in two discrete phases, then genuinely off, then a brief review in the evening of what the agents produced and what needs direction. Not eight hours of continuous presence. Intense spikes, real recovery, outputs arriving in between.

The Bridge

The reMarkable is not a productivity tool. It is a thinking tool. The distinction matters.

For me, the bridge is the reMarkable. Two hours every morning, offline, writing by hand. The thinking happens there — rough, unfiltered, not shaped for anyone else. When I am ready to act, I record a voice note and send it to the agent: what I worked out, what needs to happen. Raw thinking goes in. Structured output comes back. The agent turns what I said — sometimes haltingly, sometimes badly articulated — into a brief, a document, a plan, a draft. Not because it understood my thinking better than I did, but because it handles the structuring and formatting that would otherwise pull me out of thinking mode and into execution mode.

Your bridge will look different. It might be a voice note recorded on a walk. A written note in a regular notebook, photographed and sent. A spoken dictation straight into a chat window.

One tip that makes a significant practical difference: dictate rather than type. When you speak to an agent, you naturally give more context than when you type — sentences flow more freely, you include the reasoning not just the conclusion, and the output tends to be richer and more useful as a result. If your existing setup is not good for dictation, I use Wispr Flow on a daily basis. It works across any application and is the best tool I have found for this — it feels like thinking out loud, directly into the system.

What matters is that you have a consistent way to take the thinking that happened offline and hand it to the agent — so the gap between insight and action stays small.

I did not plan this routine. It emerged from noticing that my best ideas arrived on runs and hikes, not at a desk. The desk is where I execute. The movement is where I think. The bridge is how the two connect. Find your version of all three.

What the Bridge Actually Produces

Let me make this concrete with something I do every week.

Every time I have a significant conversation with a partner, a potential client, or someone in the market — I record it or take notes. The transcript goes into the agent. The full context of that relationship, that conversation, that opportunity: all of it is already there.

The next morning, offline, I reflect. Not on the details — the agent has the details. I think about the story. What is the real thing we want to say? What is the actual narrative that connects their situation to what we can do? What matters to them and how do we speak to that honestly? I write it down. That thinking might take twenty minutes. Sometimes less.

Then I dictate it to the agent. It already has the transcript, the branding guidelines, the best practices from the experts I most value on how to structure a compelling proposal. It takes my twenty minutes of story-thinking and turns it into a full proposal — the right length, the right structure, the right tone — within minutes.

The result is better than anything I produced before. Not marginally better. Substantially better.

I spent time early in my career at a Big Four consulting firm. I know exactly what the proposal process looks like from the inside: the hours of formatting, the version-control chaos, the gap between the idea you had in the meeting and the document that eventually went out. The idea was always clearer in the room than on the page. The hassle of production — the clicking, the adjusting, the making-it-look-right — sat between the thinking and the output, and it cost quality as much as time.

With Translation, that gap closes. The thinking is mine. I do it offline, where my thinking is actually good. The production is the agent's. It does not get tired, does not have opinions about fonts, does not need to be managed through a formatting crisis. It executes the brief I gave it — at the level of someone who has

studied every great proposal ever written and knows exactly what structure produces the best outcome for the specific type of opportunity in front of us.

This is what Level 1 looks like in the market. Everyone you are competing with now has access to AI. The ones who use it only for execution — generating content, filling templates, automating the mechanical — are getting marginal gains. The ones who use it for Translation — who bring their best thinking and use the agent to turn it into the highest-quality possible output — are competing on a different level entirely.

The thinking is the differentiator. The agent is the amplifier. Translation is the practice of making sure the thinking actually happens — and then making sure it does not get lost in production.

What Translation Is For

Consider what this means at scale.

In all developed economies, more than fifty percent of work is knowledge work. Knowledge work, when you strip it to its core, is thinking. The output of knowledge work is thought, applied to decisions and problems.

And yet, for most knowledge workers, the working day is structured around mandatory presence at a screen, in an office, for fixed hours. We have taken the one resource we need most — high-quality human thinking — and built an entire set of institutional arrangements designed to prevent it from functioning at its best.

This is one of the shifts the agentic era will force. When agents handle the execution, what remains for humans is thinking and judgment. And thinking happens best when you are not managing a screen.

Translation is the practice of creating the conditions for your best thinking — and then feeding that thinking back into the system through the dictation bridge. Not working less. Working differently. Fewer hours in execution mode. More hours in the conditions where the thinking that actually matters happens.

The system runs. Your job is to think well enough to direct it.

Your Sixth Step

Design your Translation rhythm. Not the ideal version — the version you can start this week.

Open an AI conversation and type this:

“I want to design a working rhythm that creates more space for my best thinking. I want to reduce time responding to inputs and increase time generating original thinking. Based on what you know about cognitive performance and focus, help me design a three-phase daily structure that works with my actual life. Here is what my current day looks like: [describe it honestly]. I am not looking for a perfect system — I want a starting point I can test this week.”

Let it ask you questions. Answer honestly. What emerges will not be my three-phase rhythm — it will be a version calibrated to your life, your constraints, your work.

One specific thing to try, regardless of what the conversation produces: identify one recurring time in your week when you could be genuinely offline for ninety minutes. Not “do less on your phone” — genuinely offline. No screen, no inputs. What you do with that time is less important than starting the practice of protecting it.

→ If you are within a job: Translation inside an organisation looks different. The two-hour offline morning is probably not an option on most workdays. But the principle applies: find the time in your week that is genuinely yours, protect it from inputs, and use it to think about what you are going to brief the agent to do next. Even thirty minutes of deliberate thinking, before the day starts, changes the quality of what you hand to the agent — and the quality of what comes back.

A — you saw the loop. G — you found the constraints. E — you stripped the waste. N — you nominated your deputies. T — you translated the freed hours into real life.

This is Agentic Living.

Closing: What I Found

I want to start with a story that is not about work.

June 24, 2026. An experiment was running in the background. The agent was handling what it handled. I was thinking about something else entirely.

I was thinking about two people — my housemates in Buenos Aires — who were about to host me for the last time before I left for Europe. I wanted to do something for them. Not a gift exactly. Something with a puzzle in it. Something that only made sense if you knew the word.

The word was “Infinito.”

I built a private webpage, accessible only by entering that word. Wrong answers returned error messages in Argentine slang. The correct answer opened a Vienna itinerary — a trip back to the Austrian house in Zell am See where the Agentic Living story began.

The idea was mine. The execution was the agent’s. In under an hour.

Agentic Living, it turns out, is not confined to professional life. Once you have crossed, it comes with you into everything. The gifts you give. The experiences you make possible for the people you love. The things you create not for any strategic reason but because the system is now good enough that you can.

That is the thing I found on the other side of the walk. Not just more capacity. More of myself, applied to more of my life.

The Honest Numbers

140 days. Here is what actually happened.

March: 853 messages sent. 63 sessions — 46 interactive, 17 autonomous. Median gap between messages: 3.3 minutes. Someone sitting at a keyboard, watching, correcting, redirecting in near-real-time. This was not autonomy. This was scaffolding.

By July: 1,171 autonomous sessions in a single month. The first four months combined had produced 33 autonomous sessions total.

The number surprises people. They expect the autonomy curve to mean the human does less. The opposite happened. As autonomous activity grew, my message count grew too. More autonomy did not mean less involvement. It meant higher total throughput. More ambitious work required more active direction. More background systems meant more things occasionally needing attention.

This is the honest version of Level 1. Not a transfer of work from human to machine. An expansion of total capacity — the human does more, the machine does more, and the output is larger than either could produce alone.

The tiredness at the end of a good agentic day is different from the tiredness at the end of a conventional one.

Conventional tiredness: a diffuse, heavy feeling of having been present for a long time. The sense of time spent rather than things made.

Agentic tiredness: a specific, earned exhaustion. The feeling of having done the hardest thinking available, and then having actually rested instead of grinding through low-value tasks for the remaining hours.

The second kind of tiredness is better. It is the tiredness that comes with the sense of having been the right person, doing the right thing, at the right level of engagement.

What the Attention Actually Went Toward

The end state of Agentic Living is not productivity. Not efficiency.

The end state is a life where the things that deserve your attention actually get it.

For me, that meant mountains. Long runs at altitude. Hours in coffee shops in places that take effort to reach and reward the effort with beauty. Meetings with people worth thinking alongside. Creative work done slowly and well. The ability to stay two weeks instead of two days, to wait for the weather window, to be present when the conditions are right.

In El Chaltén — a small town at the base of Cerro Fitz Roy in Argentine Patagonia, glaciers on three sides — I stayed for two weeks. Not because the work had slowed down. Because the architecture of the work no longer required me to be anywhere specific. And I chose to be there.

Most visitors stay two or three days. The logistics of getting there determine how long you can afford to be present. I stayed until I had seen everything I came to see — the Laguna Torre, the Fitz Roy viewpoint, one afternoon watching a glacier calve into a lake from thirty metres away.

Those are not experiences you can compress into a two-day visit. They require patience, flexibility, and the willingness to wait for the conditions to be right.

Both — the working mornings and the mountain afternoons — were possible because of the same decision made before I arrived. Build the system. Let it travel with you.

The record of all of it — the trails, the routes, the places this attention made possible — lives at runs.trillion-initiative.com



and rides.trillion-initiative.com



. Trek and Sea:

mountain trails and coastal rides through some of the most extraordinary terrain on Earth.

For every trail on runs.trillion-initiative.com, there is a short documentary. Built by my agent from the raw footage of my Meta AI glasses and my drone — no editor, no production team, no script. Just the footage, the data, and an agent that knew what to make of it. Each one is a record of a place I got to be in fully, because the work was handled and I was not managing a screen.

This is what the attention goes toward. Not productivity metrics. Life.

The Question Underneath the Question

The task list is infinite. Attention is finite.

That sentence is the most important thing in this book. Not a productivity insight. A philosophical position about how to live.

The conventional approach to this problem is prioritisation — deciding which items on the infinite list to address first. But prioritisation still treats the infinite list as the primary reality. You are always managing the backlog. The backlog is always winning.

Agentic Living changes the frame. The system handles the infinite list. The question becomes: what do I do with the protected hours?

That question has no universal answer. It has your answer. And finding your answer — what it is, what it actually looks like, what you would do if the execution overhead were gone — is, in some sense, the whole point of this book.

The three questions I want you to sit with before you close this:

One. What would tell you that your life has actually expanded — not that you're busier or more productive, but genuinely different? What would be specifically, visibly true?

Two. If you recovered five hours a week, what would you actually do with them — and what does your honest answer tell you about your priorities?

Three. Which tasks do you consciously not want to delegate — and is that a values decision or a control habit?

The third question is the hardest. It is the one people want to skip. It is also the one that, when answered honestly, tells you the most about what Level 1 actually means for you.

One thing to do this week, regardless of where you are in the process: write down three things you would do if you had no execution overhead this month. If you cannot name them in three minutes, the system question is not the most important one yet. The more important one is: what do I actually want?

The system is only as good as the life it serves.

The Record

Before I close, one more number.

July 21, 2026. I pulled the data from my Garmin API — not as a stunt, but as a way of marking the moment. The record of a life, already kept, retrieved in seconds.

4,280 activities logged since I started wearing the watch. 51,468 kilometres moved — 1.28 times the circumference of the Earth. 606,218 metres of elevation gained — 68.5 times the height of Everest.

Those numbers were not assembled for this book. They were already there, logged automatically, available on request. The agent retrieved them in response to a question I asked. The same way it retrieves everything — because everything was already there, already connected, already waiting to be made useful.

That is Level 1. Not magic. Not automation for its own sake. The systematic reduction of the distance between what you already have and what you could actually do with it.

What Comes After This Book

The individual story ends here. The larger one does not.

The people reading this book are almost certainly working in elastic sectors — software, research, creative work, education, consulting. When the cost of production drops in these fields, total demand expands. The opportunity grows faster than any individual workforce is displaced.

That creates an obligation. Not just a personal one.

If elastic sector workers use AI to produce the same output with less effort — if they treat agents as a productivity convenience — the economic transition gap widens. Displaced workers from inelastic sectors have nowhere to go. The social contract frays.

If elastic sector workers multiply their output genuinely and structurally — not as an initiative but as a fundamental change in how work is done — something different becomes possible.

Hermann Arnold, a Swiss entrepreneur who engaged with an early version of this work, framed it precisely: the bar is not one hundred times. A two times increase in the elastic half

mathematically compensates for the full displacement of the inelastic half. Most people who complete the workshop are already running at five to ten times their previous output within months. That is enough — if the distribution is right.

Going agentic is not just about your own attention. It is the appropriate response to the moment we are in.

The full argument is in a companion essay: The Transition Paradox. The short version: go agentic. And demand the institutions that distribute what that produces.

The Walk

I want to end where I began.

Zell am See. January 2025. Minus fifteen degrees. The research paper. The shaking.

I did not know, that night, what I had just understood. I did not have a framework for it yet, or a name for the crossing, or a hundred and forty days of data. I had a feeling I could not sit still with, and a walk in the cold.

Less than a year later, I had a different version of the same sensation. Not the shock of understanding what was now possible — I already knew that. Something else: the recognition that it was spreading. People I knew were crossing. People I met were crossing. The acceleration that had felt theoretical in January 2025 was visible and measurable by late 2025.

By February 2026, many more had already made it across. The question had shifted from whether this was possible to how many more people could be brought there.

That is why I wrote this book.

What I found, on the other side of that walk, is what this book describes.

The experiments. The failures. The sessions that ran while I slept. The morning in El Chaltén when I realised I could stay another week. The afternoon watching a glacier calve. The surprise built for two people with a single word and an hour of co-creation. The proposals that came back better than anything the full version of me with the full version of my time could have produced alone.

The walk is yours now.

You have the framework. You have the exercises. You have the proof that the crossing is learnable — 100% of the people who have walked through this in three days have made it across.

The only thing left is to take the first step.

And the only limit, on the other side, is your imagination.

Enzo Duit Buenos Aires · September 2026

A — you saw the loop. G — you found the constraints. E — you stripped the waste. N — you nominated your deputies. T — you translated the freed hours into real life.

This is Agentic Living. Now go build something.
