PRAXIS · THE SOVEREIGNTY FLAGSHIP

The Sovereignty Stack

Running real AI in your business without handing your data, your methods, or your customers to someone else's platform. The map — with the deep dives marked.

by **Adam Fischer** · Founder, Praxis

praxis-hq.ai · Theory into practice.

What's inside

A working guide, not a manifesto. Every recommendation is something you can do this week on hardware you already own.

1 The trade nobody reads

Here is how most people and most businesses are adopting AI: they find whichever tool is closest, they pour their work into it, and they never think about it again. The customer list goes in. The pricing goes in. The half-finished strategy, the client's private situation, the thing you'd never say out loud — all of it, into a box owned by someone else, governed by a policy you didn't read and can't negotiate.

It feels free. It isn't. You're paying with the one asset that actually compounds: everything you know, and everything your customers trusted you with.

I'm not here to scare you off AI. I run an entire company on it. The point of this paper is the opposite — I want you to use AI *harder* than the people around you, and keep the leverage instead of leasing it. That's the whole difference between adopting AI and surrendering to it.

> **The next decade belongs to the people who direct AI. It does not belong to the people who move into it and lose the keys.**

Sovereignty is not about paranoia or living off-grid. It's a business decision, and a simple one once you can see it: **know what lives where, know what it's worth, and know how to take it back.** That's the entire discipline. This paper gives you the framework and the stack to run it.

2 What sovereignty actually means

Data sovereignty gets talked about like it's a compliance checkbox or a privacy hobby. Neither. For an operator, sovereignty is **leverage** — the ability to walk, to switch, to say no, and to keep your advantage when the tool you depend on changes its price, its rules, or its owner.

Three things you're actually protecting:

- **Your data.** The raw material — customers, numbers, records, the private facts of your business. If a competitor could buy it, or a breach could leak it, it was never really yours to give away casually.
- **Your methods.** The prompts, the workflows, the hard-won way you do the thing. Feed them into a platform as "training feedback" and you've handed over the recipe, not just the

meal.

- **Your customers.** The relationship, and the trust that came with it. When they told you something in confidence, they didn't consent to it being processed by a vendor three layers away.

> **THE RESET**
>
> Sovereignty is a **spectrum, not a switch.** Nobody runs everything locally, and nobody should. The goal isn't to refuse the cloud — it's to make a *deliberate* choice for each piece of work instead of a lazy default. Some things should never leave your device. Some things belong in a frontier model in the cloud, and sending them there is exactly right. The skill is knowing which is which.

That's the reframe. You're not building a bunker. You're building a *stack* — a set of layers, from "never leaves this machine" to "cloud, and glad of it" — and a habit of putting each task on the right layer.

3 The three questions

Before any task touches AI, run it through three questions. This takes ten seconds and it's the entire decision engine. Everything else in this paper is just the machinery that lets you act on the answers.

Question 1 — How much would it hurt if this leaked?

Not "is it private" in the abstract. *If this exact text showed up in a competitor's inbox or a news story, what's the damage?* A blog draft: none. A client's medical situation, your customer database, an unannounced product: a lot. Rate it low, medium, or high. High-sensitivity work has no business on a shared cloud model, full stop — regardless of how good that model is.

Question 2 — What does doing it locally actually cost me?

Local isn't free either — it costs you capability and convenience. A small on-device model won't reason like a frontier one. If the task is "summarize these ten notes," local handles it fine. If it's "find the flaw in this legal argument," you may genuinely need the big model. Be honest about the gap instead of romanticizing local or defaulting to cloud.

Question 3 — Can I leave?

If this tool doubled its price tomorrow, or shut down, or changed its terms — could you take your work and go? If the answer is "no, everything's locked in there," you don't have a tool, you have a landlord. This question is so important it gets its own section (5). For now, just ask it every time.

THE 10-SECOND VERDICT

High sensitivity → keep it local, accept the capability hit.
Low sensitivity, needs real horsepower → cloud is fine, use the best model.
In between → this is where the stack earns its keep. Read on.

4 The stack, layer by layer

Picture five layers, from most sovereign to least. Every AI task lands on one of them. Your job is to push each task as far *down* (toward local) as it can go without crippling the result — and to send the rest up to the cloud *on purpose*, knowing exactly what you're trading.

Layer 0 — The data you never send

Before models, before tools: the hard line. There is information that simply does not go into any AI you don't control — cloud *or* local-but-synced. Government IDs, full account numbers, passwords, other people's private data you were entrusted with, anything a regulation covers. This layer isn't a model. It's a rule you write down once and never relitigate under deadline pressure.

YOUR LAYER-0 LIST (write it now)

Name the specific things that never touch AI in your business. Be concrete — "client SSNs," "the investor deck before launch," "anything from a sealed matter." A vague rule gets broken at 11pm. A named list holds.

Layer 1 — On-device models

This is the quiet revolution most people missed. A modern laptop — and increasingly a phone — can now run capable AI models entirely on the device, with nothing leaving the machine. Work that would have required a data center a few years ago now runs on hardware you already own, offline, private by construction.

What on-device models are genuinely good at today: summarizing, rewriting, drafting, classifying, extracting structure from messy text, answering questions about documents you feed them. What they're still weaker at: long chains of hard reasoning, niche expertise, the very frontier of capability. That's the trade — and for a huge share of daily work, it's a trade worth making, because the privacy is absolute.

DO THIS

Install one local model runner on your main computer this week (there are several free, friendly ones — you download an app, it downloads a model, you're chatting offline in ten minutes). Then

take one recurring task you currently do in the cloud and move it here. Notice two things: it works better than you expected, and nothing you typed ever left the room. *(Want the exact setup, no terminal, on Mac or Windows, with model picks and the honest capability costs? That's a whole Praxis guide of its own — **The Private Stack**. This paper is the why and the when; that one is the how.)*

Layer 2 — Your context, in files you own

Sovereignty isn't only about *where the model runs* — it's about *who owns the memory*. When you let a platform "remember" you, that memory is theirs; you can't export it, audit it, or carry it to a competitor. The sovereign alternative is almost boringly simple: keep your context in plain files on your own disk.

A short, dated document about your business, your customers (de-identified), your standards, your way of doing things — pasted into any AI at the start of a conversation — gives you tailored results from *any* model, cloud or local, while the memory itself stays yours. Switch tools and your context comes with you in a file. That's portability by design. *(This is the whole subject of a companion Praxis guide, The Context Brief — the point here is that the file lives on your machine, not theirs.)*

Layer 3 — Local retrieval (your documents, searchable by AI)

The next step up: pointing a model at your own library — contracts, notes, past reports, the accumulated knowledge of your business — and letting it answer questions grounded in *your* documents instead of the internet's average. Done locally, this is enormously powerful and completely private: the documents never upload; the model reads them where they sit.

You don't need to build this from scratch — the same local runners increasingly let you drop a folder of documents in and ask questions of it. The sovereignty win is total: your knowledge base becomes an AI you can interrogate, and it never leaves your control.

Layer 4 — The cloud, used on purpose

Now the part people expect a sovereignty paper to forbid. It doesn't. The frontier models in the cloud are extraordinary, and for the hardest reasoning you'll want them. Using them isn't surrender — *defaulting* to them without thought is. Use the cloud deliberately:

- **Sanitize first.** Strip the identifying details before you send. The model can reason about "a 54-year-old client with these numbers" without the name.

- **Turn off training on your data** where the setting exists — most serious tools now offer it. Read that one policy. It's the only one that matters.
- **Send the problem, not the database.** Paste the paragraph you need help with, not the whole file it came from.
- **Keep the output, own the input.** The result comes home to your files (Layer 2). The cloud was a consultant you hired for an hour, not a landlord you moved in with.

> **The cloud is a tool you reach for, not a place you live. That one distinction is 80% of sovereignty in practice.**

5 The exit test

Everything above collapses into a single question, and it's the truest measure of whether you're sovereign or captured: **can you leave?**

Not "will you." Can you. If your best prompts live only inside one company's interface, if your context exists only as their "memory," if your documents were uploaded and never kept locally — then leaving means starting over, which means you'll never leave, which means you have no leverage, which means the price and the terms are theirs to dictate. That's not a partnership. That's a lease with no exit.

The sovereign setup passes the exit test by construction:

Asset	Captured (can't leave)	Sovereign (portable)
Your prompts & methods	Saved inside one platform	Plain text files on your disk
Your context / "memory"	The vendor's memory feature	A dated file you paste anywhere
Your knowledge base	Uploaded, not retained locally	Local folder, searchable in place
Your daily driver model	The only one you can use	Any model — you A/B them freely

Run the test on your current setup right now. Wherever the answer is "captured," that's your next weekend project. You don't have to fix all four this month. You just have to stop adding to the captured column. *(For a full afternoon-long inventory of where your AI life actually lives — every platform, every setting, export and true deletion — the Praxis guide **The Sovereignty Audit** walks the whole sweep.)*

6 Build your own stack this week

Enough theory. Here's the practical starter — five moves, none of which take more than an evening, in the order I'd do them.

1. **Write your Layer-0 list.** Ten minutes. The named list of what never touches AI. Tape it up.
2. **Install one local model runner.** One evening. Now you have a private AI that works offline for everything on the low end of the sensitivity scale.
3. **Move one recurring task to local.** This week. Prove to yourself the capability gap is smaller than the fear.
4. **Start a context file.** One page, dated, on your own disk. Paste it into every AI conversation. Watch generic answers become yours — from any model.
5. **Run the exit test** on everything else, and pick the one captured asset that would hurt most to lose. That's next month's project.

THE HONEST BOTTOM LINE

You will still use the cloud. You should. The difference is that after this week, the cloud is one deliberate layer in a stack you control — not the whole building with your name on someone else's lease. Sovereignty isn't all-or-nothing. It's a habit of putting each task on the right layer, and never letting the captured column grow.

7 The worksheet

Copy this into a note and keep it where you work. Run a task through it whenever you're about to reach for AI on anything that matters.

```
# THE SOVEREIGNTY WORKSHEET

TASK: _____________________________________________

1. LEAK TEST — if this exact text leaked, the damage is:
   [ ] None/low   [ ] Medium   [ ] High

2. HORSEPOWER — does this need frontier-level reasoning?
   [ ] No, a small local model can do it
   [ ] Maybe   [ ] Yes, genuinely

3. VERDICT (from 1 + 2):
   High leak → LAYER 0 (never) or LAYER 1 (local only)
   Low leak + needs power → LAYER 4 (cloud, sanitized)
   Everything else → push it DOWN as far as it goes

4. IF CLOUD, BEFORE I SEND:
   [ ] Stripped identifying details
   [ ] Training-on-my-data is OFF
   [ ] Sending the problem, not the whole file
   [ ] Output comes home to my own files

# MY LAYER-0 LIST — never touches any AI
- _____________________________________________
- _____________________________________________
- _____________________________________________

# MY EXIT TEST — can I leave? (fix the "captured" ones)
Prompts & methods ..... [ ] portable  [ ] captured
Context / memory ....... [ ] portable  [ ] captured
Knowledge base ......... [ ] portable  [ ] captured
Daily model ............ [ ] portable  [ ] captured
```

8 Where this goes

Start here and something shifts. AI stops being a place you visit and hand things to, and becomes a set of tools you point at your own work on your own terms. You use it more, not less — because you finally trust where the sensitive stuff goes. You stop flinching at the cloud and start using it surgically. And the day a tool changes its price or its rules, you shrug, because you can leave.

That's the whole game. Not refusing the future — *directing* it, and keeping the keys.

> **Adopt AI without surrender. Know what lives where, what it's worth, and how to take it back.**

This is the map. The Praxis library has the deep dives, each one a layer of this stack in full:

- **The Private Stack** — the no-terminal setup for running AI on your own machine (Layer 1).
- **The Context Brief** — the one-page file that makes any model yours, and stays yours (Layer 2).
- **The Sovereignty Audit** — the afternoon sweep of where your AI life lives, and taking it back (the exit test, in full).

Start with the layer that would hurt most to lose. Theory into practice, one layer at a time.
