

A TALK ON PERSONAL AI SYSTEMS

Principles of a personal operating system.

The tools are the least interesting layer. Eight principles from seven months of building a personal AI system, and a look inside mine.

Isabella Isotta

BEFORE ANYTHING ELSE

Claude is my _____.

Finish the sentence. One word only.

WHY THE WORD MATTERS

A tracker is not a butler, and a butler is not a thought partner. The word you pick says a lot about what you have actually built - and what you haven't.

THE ONE THING I WANT TO SHIFT

Where the leverage actually lives.

WHAT I THOUGHT WHEN I STARTED

- Using AI well means **picking the right tools**
- and **writing good prompts**
- so the skill to learn is prompting, and the decision to make is which subscription to buy

WHAT I THINK NOW

- The tools are the least interesting layer - the leverage is in the **system you design around how you think**
- This year's word for it is the harness - and everyone's building them for companies. **This talk is about building your own**
- A personal harness is made, not picked - and the design principles are learnable

Extend, replace, or gap-fill?

Every piece of a setup does one of three things to a faculty you already have. Only one of them costs you anything.

Extend

The thinking still runs through your head - you just get further with it. *An essay draft comes back with the three notes in my own vault that argue against it. The argument is still mine to make.*

Gap-fill

It does something you never could. **Pure gain.** *A morning briefing that has read forty sources before I'm awake. No version of me was doing that.*

Replace

It does the thinking and you collect the output. The faculty **atrophies**. *Summarising: hand every long read to a model and the muscle for reading a hard argument start to finish goes quietly.*

WHICH LOSSES ACTUALLY MATTER

Retiring a faculty on purpose is legitimate. The test - **is it still owed of you unassisted**, in rooms the system cannot enter, **and was it strong before?** Both yes, protect it. Otherwise delegate without guilt.

A 20-SECOND DETOUR INTO PHILOSOPHY

A thought experiment.

Clark and Chalmers, 1998. Inga hears about an exhibition, remembers the museum is on 53rd Street, and sets off. Otto has Alzheimer's, keeps every address in a notebook he carries everywhere and trusts completely, looks up the same street, and sets off.

THE QUESTION

Inga's belief lived in her head. Otto's lived in the notebook. Both walked to the same door for the same reason – so what, exactly, is the difference between where their minds stop?

ONE PICTURE BEFORE THE PRINCIPLES

The whole thing, in four layers.

The substrate

Plain markdown files in flat numbered folders, in a git repository on my machine - every note, essay, lecture, person and task. Pushed to a private mirror every 15 minutes, without my touching it.

The machinery

Scripts and scheduled jobs that move things deterministically - transcription, filing, syncing, backup. No model involved, because none is needed.

The intelligence

Claude Code reading the same files: **skills** for repeat jobs, **agents** for the judgment calls, and **hooks** that check its work on every write.

The connections

MCP servers into mail, calendar and files, plus a browser for anything behind a login. All swappable, and deliberately the easiest part to tear out.

Live: the vault open in Obsidian, and the same folder in Terminal - no database anywhere.

1 · Own the memory, rent the reasoning.

THE PRINCIPLE

The durable half of the system is memory - notes, history, corrections, structure - and it lives in plain text on your machine. The reasoning is compute you rent by the month, and it should be swappable without touching a single file. Never let the thing that thinks own the thing that remembers.

- Markdown, folders and git are fully portable. **The record itself has no lock-in at all.**
- MCP is an open standard, so the integrations mostly travel between tools too.
- The tool-specific glue - hooks, permissions, skills - took real time to build, and I won't pretend it rebuilds itself. But it is mine, it is readable, and nobody else holds it over me.

2 · Hard structure, fluid meaning.

BE RIGID ABOUT

- The **folder skeleton** - flat, numbered, boring
- One **metadata schema** for every note
- **Naming rules** that never bend

BE LOOSE ABOUT

- The **areas of your life** - they change, and the system should let them
- **What connects to what** - links and metadata do this, so meaning can move without files moving
- What anything **means this month** versus last

3 · Every step is deterministic or judgment – on purpose.

- Every step in a pipeline is either a **script** (transcription, filing, linting - things that must never vary) or a **judgment call** (summarising, deciding - where the right answer depends on context). Knowing which is the whole discipline.
- Instructions in a context file are advisory - the model will eventually ignore one. **A rule you care about becomes a check that runs on every write and fails loudly.**

SOME OF MINE

A metadata check that runs on every single write and fails loudly. A voice gate that runs before anything can leave in my name, checking it against my own writing. A check that fires the moment a new person note appears. None of them asks permission, and none of them depends on anyone remembering.

4 · Tell it your goals, not your spec.

THE PRINCIPLE

"Make me a study plan" gets you a study plan. "I want to stay on top of coursework without giving up my evenings" gets you a much better one. Both are instructions – the difference is altitude. Say what you are optimising for, and let it choose the artefact.

Standing missions

The handful of things the system always holds - what this year is actually for. Loaded every session, never restated.

The goals of this block

What I'm optimising for right now, this term. Changes a few times a year.

The goal of this session

Stated at the top of the hour. This is the one people skip, and it's the one that stops a session drifting.

Because all three are explicit, it can trade them off - and push back when a request works against something bigger.

Live: the goals file loading at session start, and the model pushing back on a request because of it. The one idea I properly copied - Daniel Miessler's PAI.

5 · Build for how you think.

THE PRINCIPLE

If the system is part of your mind at all, it has to be shaped like yours. The bottom of the stack is engineering and is the same for everyone – but the interface, how it surfaces things to you and how you feed it, is the part nobody else can design for you.

- Build it **flexible enough to bend to how you already work**, not rigid enough that you have to bend to it. That flexibility is where the leverage is – a system fighting your habits loses.
- I killed urgency scoring in my own task system because I need the full map in my head in advance – a friend of mine runs the exact opposite, and both of us are right.
- Then the test from slide three, unchanged: **is the faculty it touches still owed of me unassisted, in a room this system cannot enter?** If it is, build it to extend rather than replace.

6 · Close the loop - never give the same feedback twice.

THE PRINCIPLE

*Every correction should become either a **permanent memory** or a **mechanical check**. The test is whether you have ever told it the same thing twice.*

- When I flag a phrase I hate in a draft, it goes onto a banned list that a script checks every outgoing email against - I see it once, and never again.
- **149 memories** so far: how I work, what a project is for, rulings I've made once and never repeated.
- Writing everything into memory isn't the answer either - an unbounded list is just a longer thing to read. Knowing which corrections deserve to be permanent is the craft.

7 · Scale autonomy with verifiability.

THE PRINCIPLE

Generation is cheap - your attention is the scarce resource in the loop. So every delegated surface gets a permission boundary (what it structurally cannot do) and a verification surface sized to the blast radius (what you check, and how fast).

- Mine is scoped to a draft-only mail surface, so there is no send button to misfire. It never bulk-edits without piloting one file first, and it proposes changes to anything important and waits.
- None of that is distrust - it's what lets me hand over more each month, because checking stays fast.

8 · Separate what appreciates from what depreciates.

APPRECIATES AS MODELS IMPROVE

- Your **data** - notes, history, everything captured
- Your **goals and feedback** - every correction you have ever given it
- Your **structure and taste**
- Same model, richer record, better answers

DEPRECIATES AS MODELS IMPROVE

- **Prompts** - useful, but built to be rewritten
- **Scaffolds and glue code**
- Keep this layer thin, separate and cheap to replace
- When the next model lands, you should already know which files you'd touch

THE PART PEOPLE ACTUALLY ASK ABOUT

It doesn't just serve you - it extends you.

A bench of advisors

Seven agents, each grounded in a corpus I built - **every essay Paul Graham has published**, the LessWrong Sequences, my negotiation course - which answer by quoting the source verbatim, never by impersonating its author.

A voice gate

Every piece of outgoing writing is checked against my own unassisted writing, including a growing banned-phrases list of things I've caught AI saying in my name.

A relationship record

264 people and 17 companies, where every meeting note, email and task involving someone assembles itself on their page without my touching it.

A taste record

The one I use most, which probably tells you something. I have always been a massive lister. Now the lists make themselves. I speak a verdict on whatever I have read or watched into my phone, and it files itself.

Live: a pre-run consultation, and the list of my own banned phrases.

A good system makes new areas of life cheap.

THE TEST I'D GIVE ANY SETUP

What is the marginal cost of adding a whole new area of your life? When Wharton arrived - new courses, a new platform, new readings and deadlines - I didn't build a Wharton system. I asked what the existing design already handled, and the answer was almost everything.

ALREADY THERE

- Lectures, readings and deadlines flow through the **existing pipelines**
- The **same schema** organises all of it

ACTUALLY NEW

- One **Canvas sync** - a new source
- One **map page** - a new view over existing structure
- That's the whole build

FOR THE WAY HOME

Where the ideas came from.

Andrej Karpathy

The llm-wiki idea - raw sources, a persistent markdown wiki the model maintains, Obsidian as the human's window onto it.

Daniel Miessler

Personal AI Infrastructure (PAI) - the most complete open framework, and where my goals-files idea comes from.

Simon Willison

The clearest writing on what agents can and can't be trusted with - read him before you connect anything to your email.

Steph Ango

"File over app" - the case for owning your data as plain files. The whole substrate argument in about a page.

Ethan Mollick

Co-Intelligence - the book on working with AI day to day.

And I write about self and system at I Heart My Strange Loop.