danielmiessler / **Personal_AI_Infrastructure** Public

Code | Issues 81 | Pull requests 95 | Discussions | Actions | Projects | Security and quality | Insights

main | 3 Branches | 19 Tags | Go to file | Go to file | Code

danielmiessler README: subtle Life OS framing + canonical RIoT URL dc05fe9 · yesterday

.claude	Revert "PAI v4.0.0 — Lean and Mean"	2 months ago
.github	Add Claude Code GitHub Workflow (#938)	last month
Packs	feat: Add PAI Packs icon (PP block style, tr…	last month
Releases	Fix Pi repo URL to badlogic/pi-mono (from …	3 weeks ago
Tools	Revert "PAI v4.0.0 — Lean and Mean"	2 months ago
images	Revert "PAI v4.0.0 — Lean and Mean"	2 months ago
.env.example	Added 2.3 Release to new Releases direct…	3 months ago
.gitattributes	fix: Add .gitattributes to enforce LF line endi…	4 months ago
.gitignore	feat: Add intelligent PAI update system with…	4 months ago
.pai-protected.json	Update .pai-protected.json: add 4 new secu…	last week
LICENSE	Enhanced README with PAI video insights…	7 months ago
PLATFORM.md	Revert "PAI v4.0.0 — Lean and Mean"	2 months ago
README.md	README: subtle Life OS framing + canonic…	yesterday
SECURITY.md	Revert "PAI v4.0.0 — Lean and Mean"	2 months ago

README | MIT license | Security

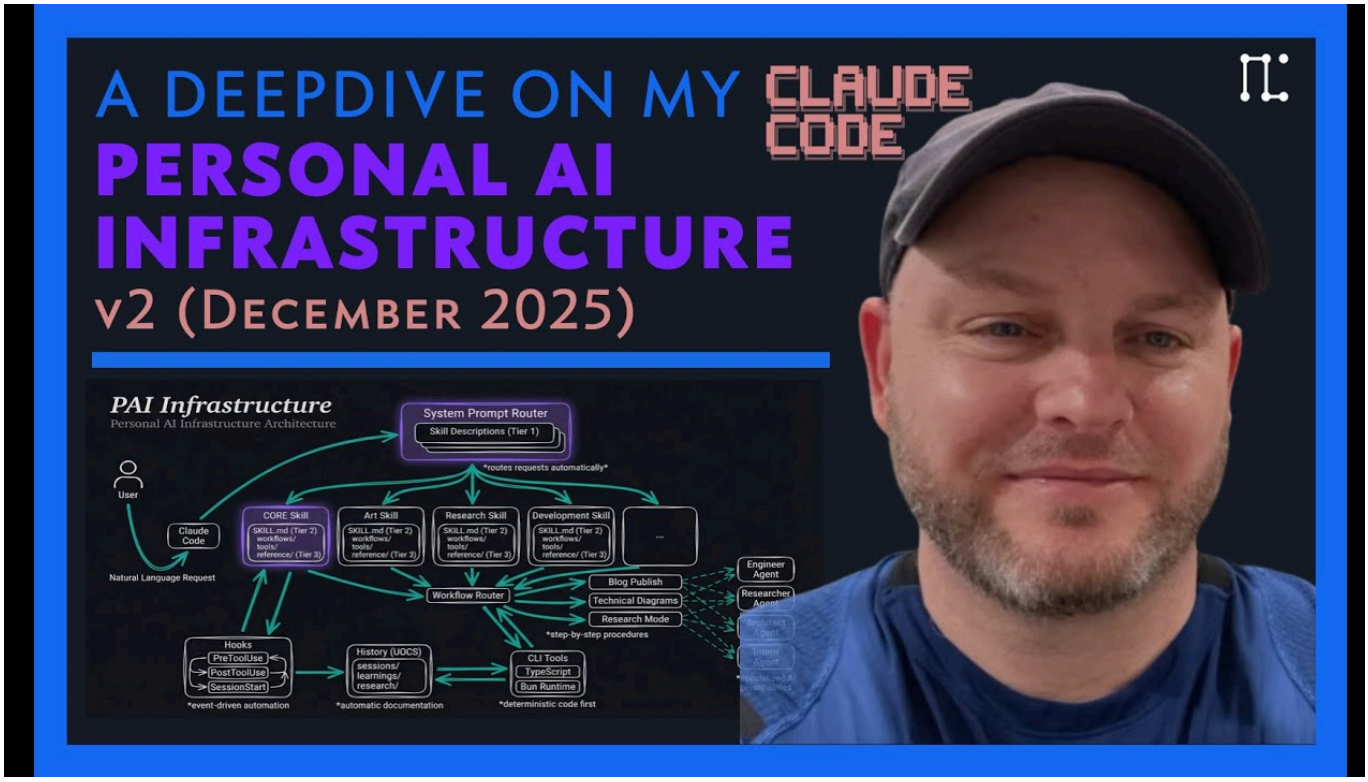
Personal AI Infrastructure

Everyone needs ac

Overview: Purpose · What is PAI? · New to AI? · Principles · Primitives

Get Started: Installation · Releases · Packs

Resources: FAQ · Roadmap · Community · Contributing



Watch the full PAI walkthrough | Read: The Real Internet of Things

> Important
>
> **PAI v4.0.3 Released** — 3 patch updates since v4.0.0 with 30+ community-contributed fixes: Linux compatibility, JSON parsing, installer improvements, portability, and upgrade migration.
>
> Release notes → | All releases →

AI should magnify everyone—not just the top 1%.

The Purpose of This Project

PAI exists to solve what I believe is the P0 problem in the world:

Only a tiny fraction of humanity's creative potential is activated on Earth.

Most people don't believe they have valuable contributions to make. They think there are "special" people—and they aren't one of them. They've never asked who they are, what they're about, and have never articulated or written it down. This makes them catastrophically vulnerable to AI displacement. Without activation, there is no high-agency.

So our goal with PAI is to activate people.

PAI's mission is twofold:

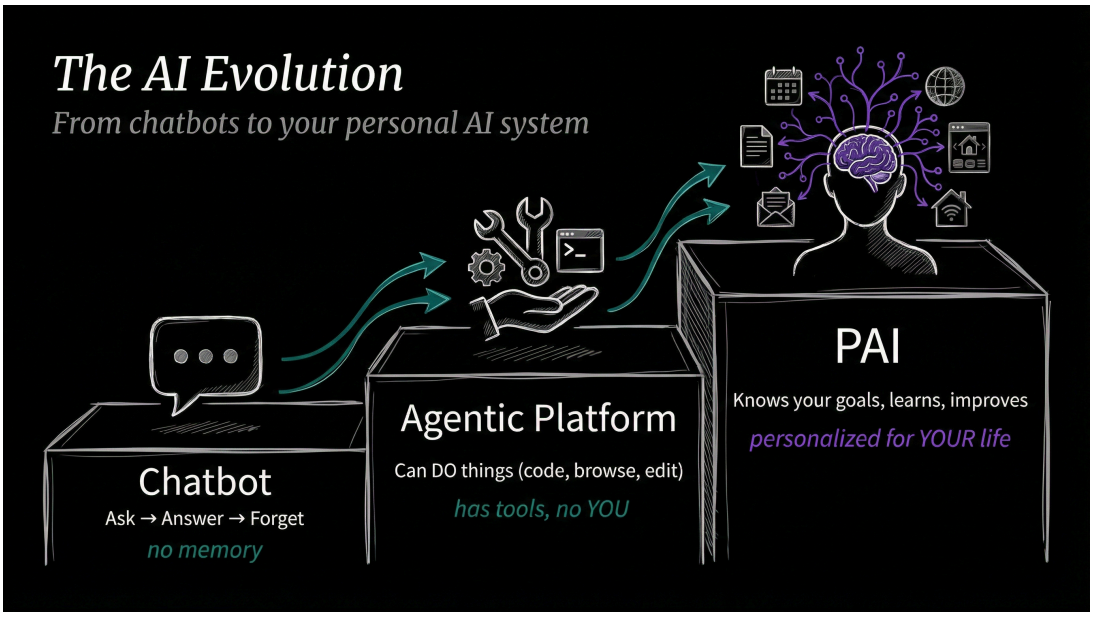
1. **Activate as many people as possible** — Help people identify, articulate, and pursue their own purpose in life through AI-augmented self discovery
2. **Make the best AI available in the world accessible to everyone** — Ensure this quality of AI infrastructure isn't reserved for just the rich or technical elite.

That's why this is an open-source project instead of private.

New to This? Start Here

You've probably used ChatGPT or Claude. Type a question, get an answer. Simple.

You can think of AI systems as **three levels**:



Chatbots

ChatGPT, Claude, Gemini—you ask something, it answers, and then it forgets everything. Next conversation starts fresh. No memory of you, your preferences, or what you talked about yesterday.

The pattern: Ask → Answer → Forget

Agentic Platforms

Tools like Claude Code. The AI can actually *do* things—write code, browse the web, edit files, run commands.

The pattern: Ask → Use tools → Get result

More capable, but it still doesn't know *you*—your goals, your preferences, your history.

PAI (Personal AI Infrastructure)

Now your DA **learns and improves**:

- **Captures every signal** — Ratings, sentiment, verification outcomes
- **Learns from mistakes** — Failures get analyzed and fixed

- **Gets better over time** — Success patterns get reinforced
- **Upgrades itself** — Skills, workflows, even the core behavior evolves

Plus it knows:

- **Your goals** — What you're working toward
- **Your preferences** — How you like things done
- **Your history** — Past decisions and learnings

The pattern: Observe → Think → Plan → Execute → Verify → **Learn** → Improve

The key difference: **PAI learns from feedback**. Every interaction makes it better at helping *you* specifically.

What is PAI?

PAI is a Personalized AI Platform designed to magnify your capabilities.

It's designed for humans most of all, but can be used by teams, companies, or Federations of Planets desiring to be better versions of themselves.

The scale of the entity doesn't matter: It's a system for understanding, articulating, and realizing its principal's goals using a full-featured Agentic AI Platform.

Who is PAI for?

Everyone, full stop. It's the anti-gatekeeping AI project.

- **Small business owners** who aren't technical but want AI to handle invoicing, scheduling, customer follow-ups, and marketing
- **Companies** who want to understand their data, optimize operations, and make better decisions
- **Managers** who want to run their teams more effectively—tracking projects, preparing for reviews, and communicating clearly
- **Artists and creatives** who want to find local events, galleries, and opportunities to showcase their work
- **Everyday people** who want to improve their lives—better fitness routines, stronger social connections, personal finance, or just getting organized
- **Developers** using AI coding assistants who want persistent memory and custom workflows
- **Power users** who want their AI to know their goals, preferences, and context
- **Teams** building shared AI infrastructure with consistent capabilities
- **Experimenters** interested in AI system design and personal AI patterns

What makes PAI different?

The first thing people ask is:

> How is this different from Claude Code, or any of the other agentic systems?

Most agentic systems are built around tools with the user being an afterthought. They are also mostly task-based instead of being goal-based using all the context available to them. PAI is the opposite.

Three core differentiators:

1. **Goal Orientation** — PAI's primary focus is on the human running it and what they're trying to do in the world, not the tech. This is built into how the system executes all tasks.
2. **Pursuit of Optimal Output** — The system's outer loop and everything it does is trying to produce the exact right output given the current situation and all the contexts around it.
3. **Continuous Learning** — The system constantly captures signals about what was done, what changes were made, what outputs were produced for each request, and then how you liked or disliked the results.

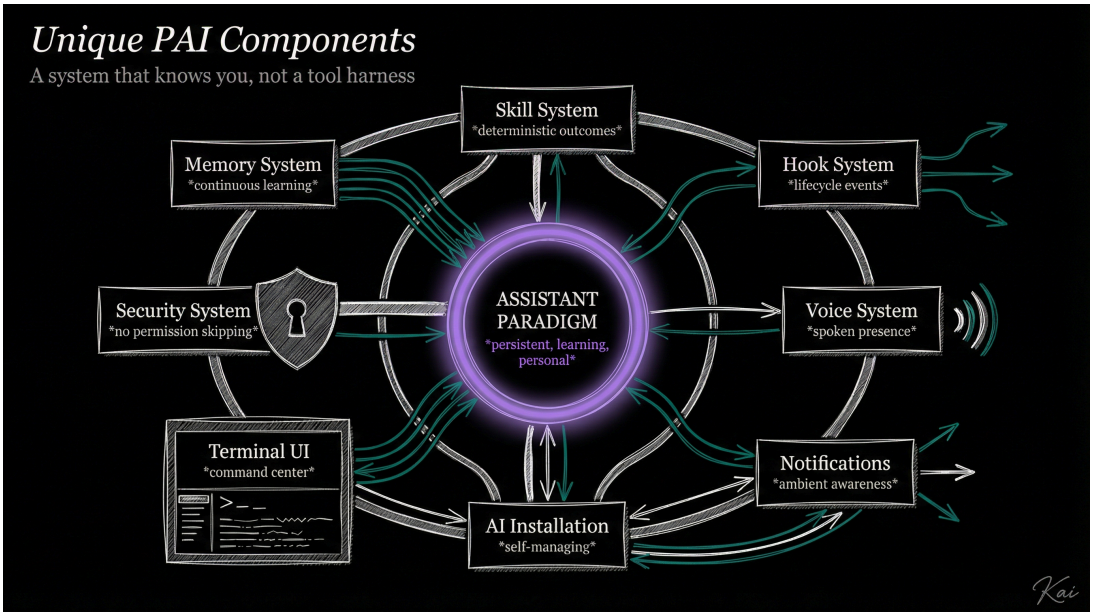
The PAI Principles

These principles guide how PAI systems are designed and built. **Full breakdown →**

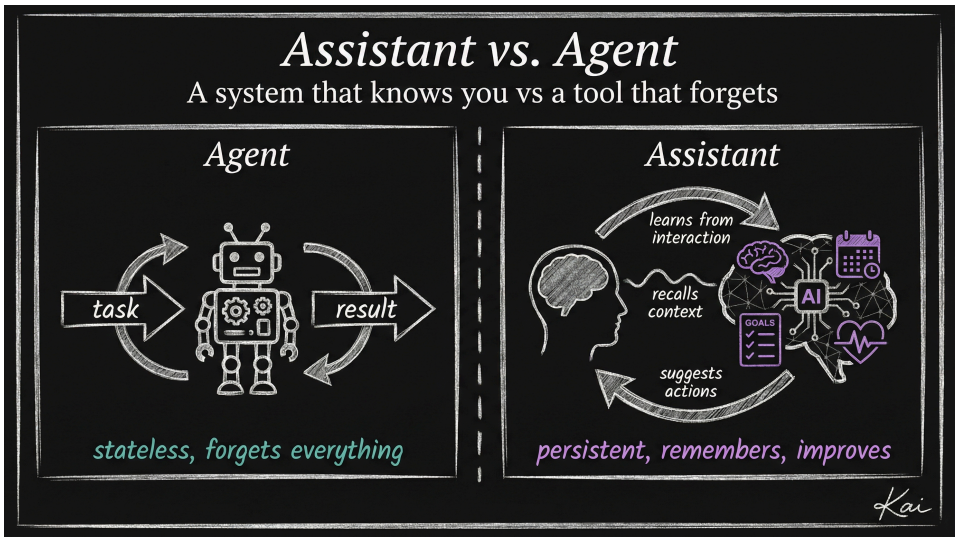
#	Principle	Summary
1	**User Centricity**	PAI is built around you, not tooling. Your goals, preferences, and context come first—the infrastructure exists to serve them.
2	**The Foundational Algorithm**	The scientific method as a universal problem-solving loop: Observe → Think → Plan → Build → Execute → Verify → Learn. Define the ideal state, iterate until you reach it.
3	**Clear Thinking First**	Good prompts come from clear thinking. Clarify the problem before writing the prompt.
4	**Scaffolding > Model**	System architecture matters more than which model you use.
5	**Deterministic Infrastructure**	AI is probabilistic; your infrastructure shouldn't be. Use templates and patterns.
6	**Code Before Prompts**	If you can solve it with a bash script, don't use AI.
7	**Spec / Test / Evals First**	Write specifications and tests before building. Measure if the system works.
8	**UNIX Philosophy**	Do one thing well. Make tools composable. Use text interfaces.
9	**ENG / SRE Principles**	Treat AI infrastructure like production software: version control, automation, monitoring.
10	**CLI as Interface**	Command-line interfaces are faster, more scriptable, and more reliable than GUIs.
11	**Goal → Code → CLI → Prompts → Agents**	The decision hierarchy: clarify goal, then code, then CLI, then prompts, then agents.
12	**Skill Management**	Modular capabilities that route intelligently based on context.
13	**Memory System**	Everything worth knowing gets captured. History feeds future context.
14	**Agent Personalities**	Different work needs different approaches. Specialized agents with unique voices.
15	**Science as Meta-Loop**	Hypothesis → Experiment → Measure → Iterate.
16	**Permission to Fail**	Explicit permission to say "I don't know" prevents hallucinations.

PAI Primitives

While the Principles describe the *philosophy* of PAI, the Primitives are the *architecture*—the core systems that make everything work.

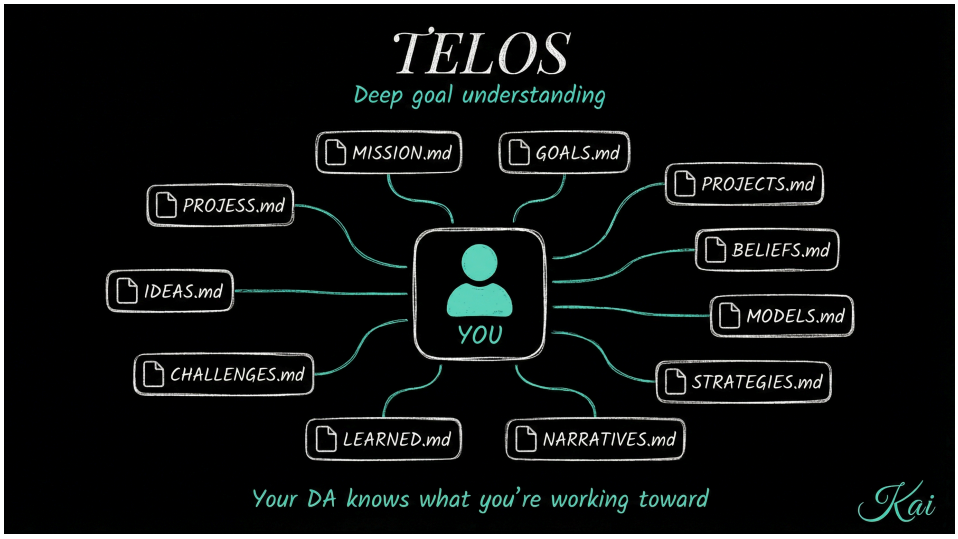


These primitives work together to create the experience of working with a system that understands and knows you—as opposed to a tool harness that just executes commands.



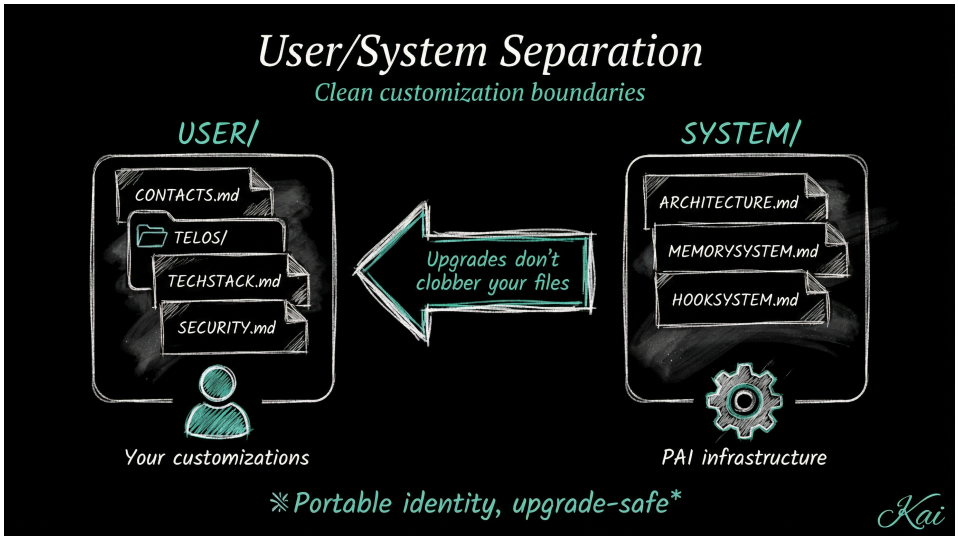
Assistant vs. Agent-Based AI Interaction

PAI treats AI as a persistent assistant, friend, coach, and mentor rather than a stateless agent that runs tasks. An assistant knows your goals, remembers your preferences, and improves over time. An agent executes commands and forgets.



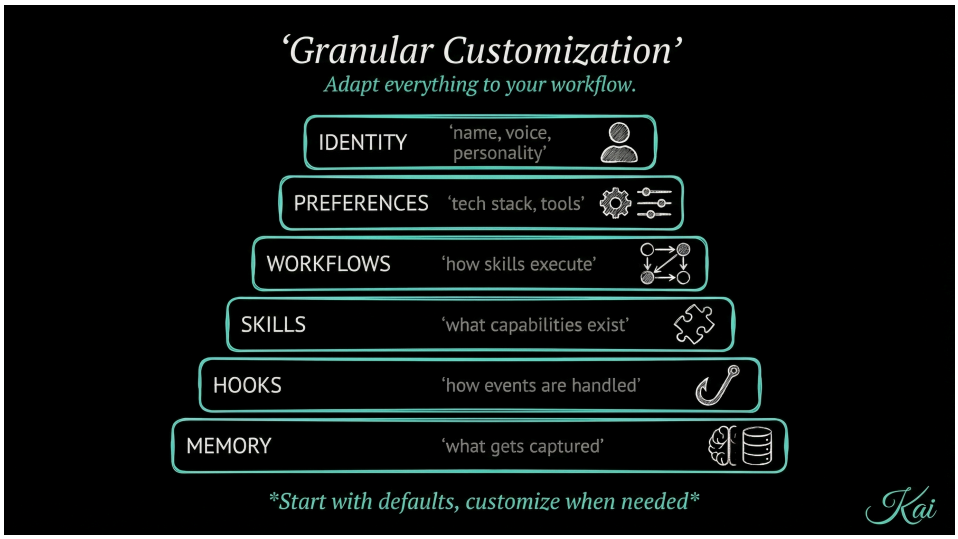
TELOS (Deep Goal Understanding)

10 files that capture who you are: MISSION.md, GOALS.md, PROJECTS.md, BELIEFS.md, MODELS.md, STRATEGIES.md, NARRATIVES.md, LEARNED.md, CHALLENGES.md, IDEAS.md. Your DA knows what you're working toward because it's all documented.



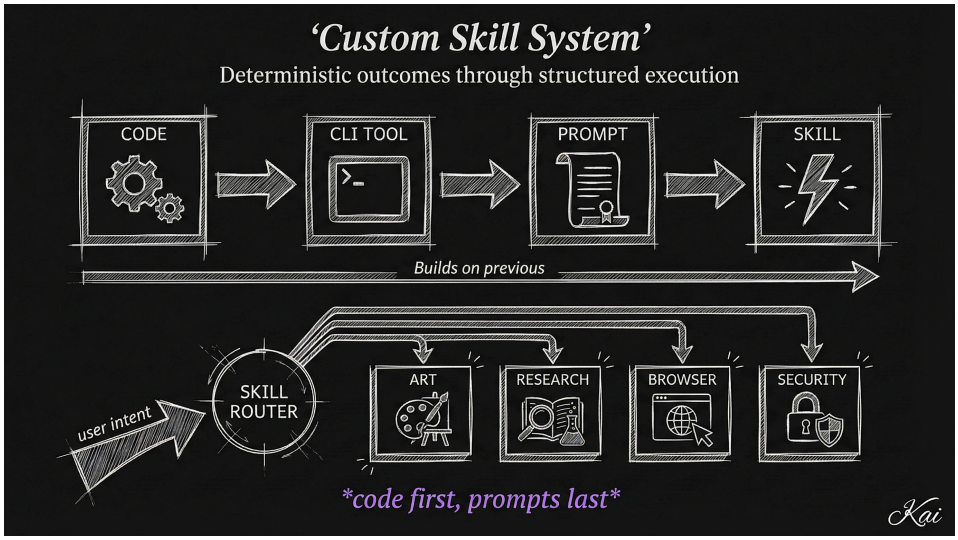
User/System Separation

Your customizations live in USER/. PAI infrastructure lives in SYSTEM/. When PAI upgrades, your files are untouched. Portable identity, upgrade-safe.



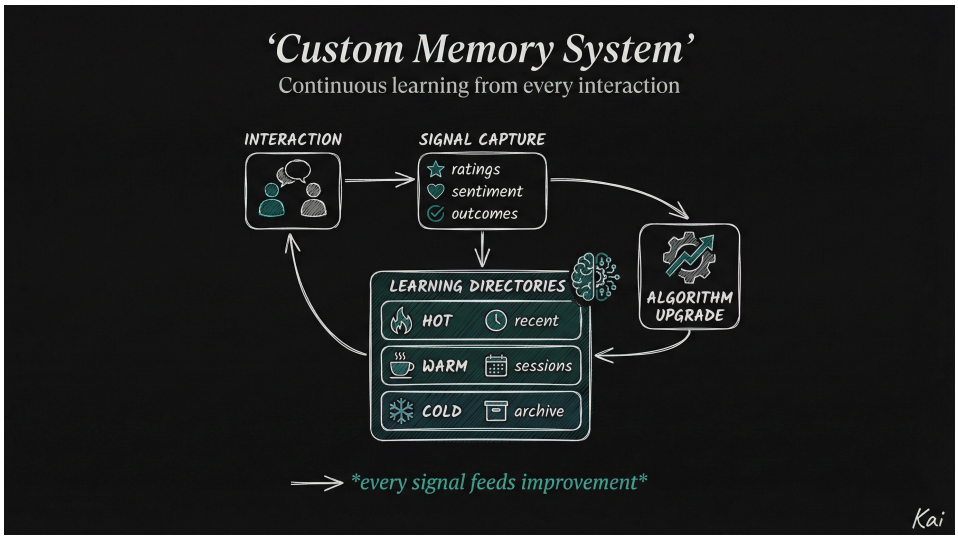
Granular Customization

Six layers of customization: Identity (name, voice, personality), Preferences (tech stack, tools), Workflows (how skills execute), Skills (what capabilities exist), Hooks (how events are handled), and Memory (what gets captured). Start with defaults, customize when needed.



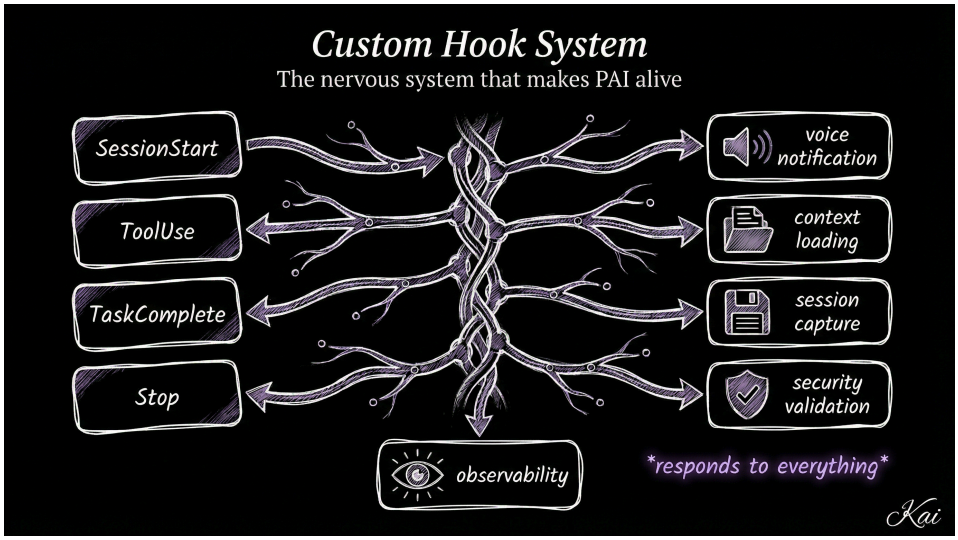
Skill System

Highly focused on consistent results. It has a structure that puts *deterministic outcomes first* by going from CODE -> CLI-BASED-TOOL -> PROMPT -> SKILL instead of a haphazard structure.



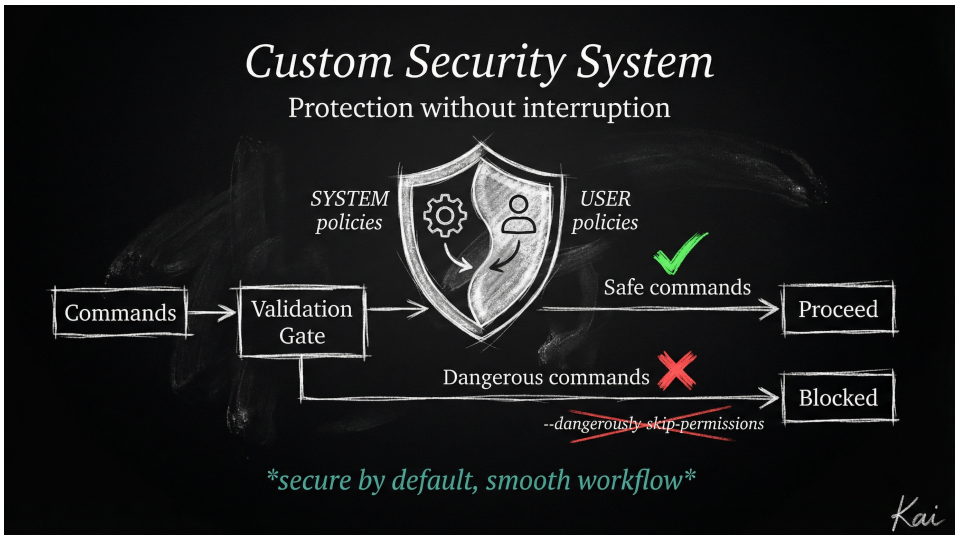
Memory System

Focused on continuous learning. Every interaction generates signals—ratings, sentiment, successes, failures—that feed back into improving the system. Three-tier architecture (hot/warm/cold) with phase-based learning directories.



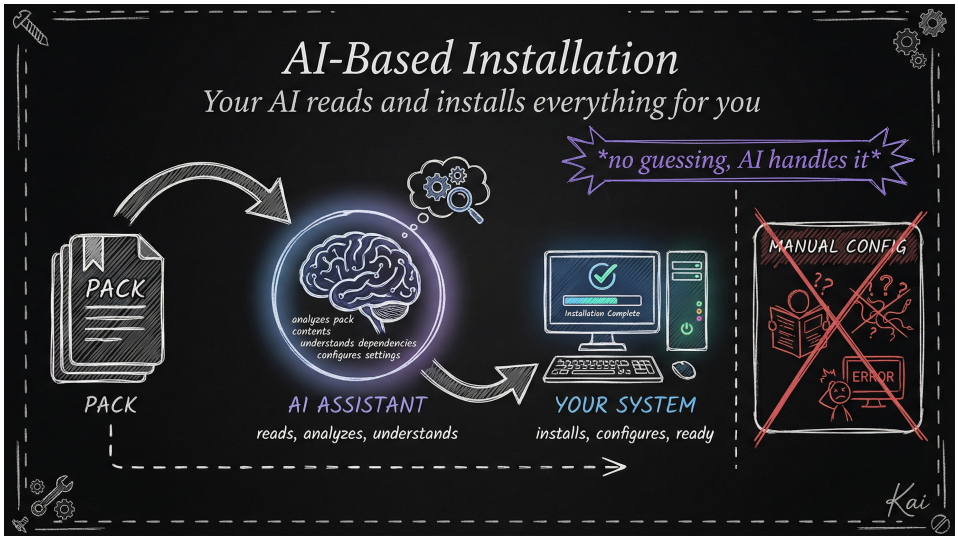
Hook System

Responds to lifecycle events—session start, tool use, task completion, and more. 8 event types enable voice notifications, automatic context loading, session capture, security validation, and observability.



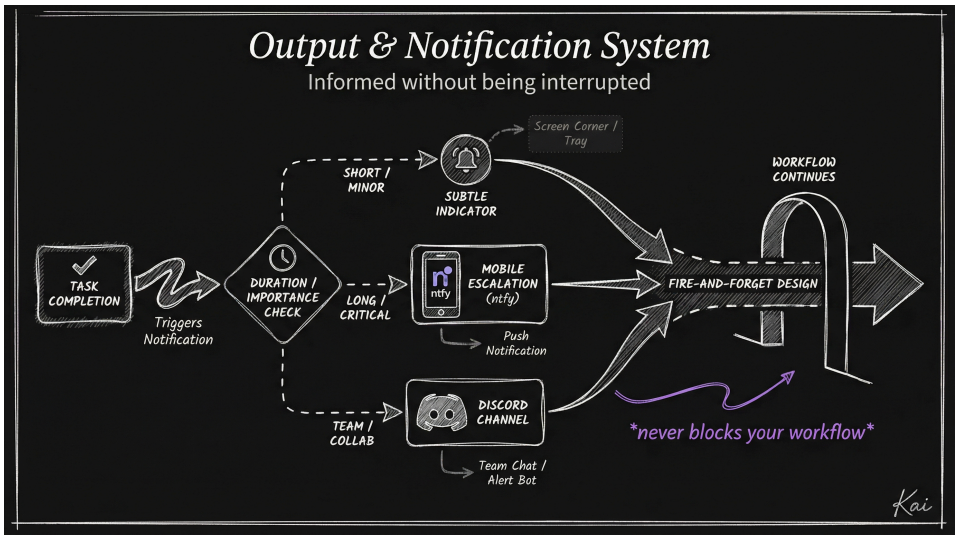
Security System

Defines system and user-level security policies by default. You don't have to run with `--dangerously-skip-permissions` to have an uninterrupted experience. PAI's security hooks validate commands before execution, blocking dangerous operations while allowing normal workflows to proceed smoothly.



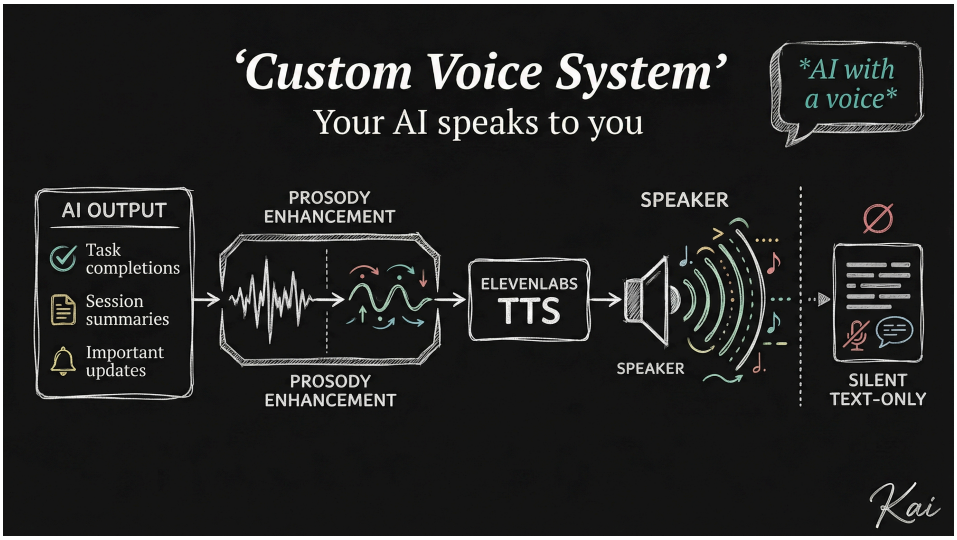
AI-Based Installation

The GUI installer handles everything—prerequisites, configuration, and setup. No manual configuration, no guessing.



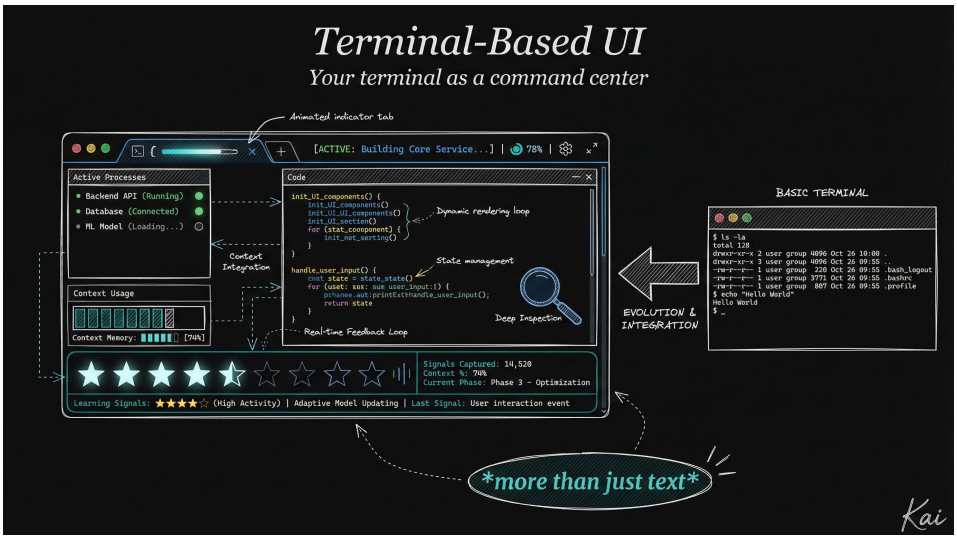
Notification System

Keeps you informed without being intrusive. Push notifications via ntfy for mobile alerts, Discord integration for team updates, and duration-aware routing that escalates for long-running tasks. Fire-and-forget design means notifications never block your workflow.



Voice System

Powered by ElevenLabs TTS. Hear task completions, session summaries, and important updates spoken aloud. Prosody enhancement makes speech sound natural. Your AI has a voice.



Terminal-Based UI

Rich tab titles and pane management. Dynamic status lines show learning signals, context usage, and current task state. Your terminal is a command center.

🚀 Installation

> Caution
>
> **Project in Active Development** — PAI is evolving rapidly. Expect breaking changes, restructuring, and frequent updates. We are working on stable and development branches, but currently it's all combined.

Fresh Install

```
# Clone the repo
git clone https://github.com/danielmiessler/Personal_AI_Infrastructure.git
cd Personal_AI_Infrastructure/Releases/v4.0.3
```

```
# Copy the release and run the installer
cp -r .claude ~/ && cd ~/.claude && bash install.sh
```

The installer will:

- Detect your system and install prerequisites (Bun, Git, Claude Code)
- Ask for your name, AI assistant name, timezone, and temperature unit preference
- Clone/configure the PAI repository into `~/.claude/`
- Set up voice features with ElevenLabs (optional)
- Configure your shell alias and verify the installation

After installation: Run `source ~/.zshrc && pai` to launch PAI.

Upgrading from a Previous Version

```
# 1. Back up your current installation
cp -r ~/.claude ~/.claude-backup-$(date +%Y%m%d)

# 2. Clone and copy the new release over your installation
git clone https://github.com/danielmiessler/Personal_AI_Infrastructure.git
cd Personal_AI_Infrastructure/Releases/v4.0.3
cp -r .claude ~/

# 3. Run the installer (detects existing installation, preserves your data)
cd ~/.claude && bash install.sh

# 4. Rebuild your CLAUDE.md
bun ~/.claude/PAI/Tools/BuildCLAUDE.ts
```

> Tip
>
> The installer **auto-detects** existing installations. It preserves your `USER/` files, merges `settings.json` (only updating installer-managed fields like identity and version), and never overwrites your hooks, statusline, or custom configuration.

Post-upgrade checklist:

- [ ] Verify your identity in `settings.json` (name, AI name, timezone)
- [ ] Confirm the statusline displays correctly
- [ ] Test voice notifications (if enabled)
- [ ] Run a simple prompt to confirm PAI responds correctly

📦 PAI Packs

Don't want to install all of PAI? **Packs** are standalone, AI-installable capabilities you can add one at a time. Each pack is self-contained — your AI reads the install guide and sets everything up for you. No PAI installation required.

Point your AI at any pack and say "install this":

```
"Install the Research pack from PAI/Packs/Research/"
```

Your AI walks through a 5-phase wizard: system analysis, user questions, backup, installation, verification.

Available Packs

Pack	What It Does
ContextSearch	`/context-search` and `/cs` — instant recall of prior work sessions
Agents	Custom agent composition from traits, voices, and personalities
ContentAnalysis	Wisdom extraction from videos, podcasts, articles, and YouTube

Pack	What It Does
Investigation	OSINT and investigation — company intel, people search, domain lookup
Media	AI image generation, diagrams, infographics, and Remotion video
Research	Multi-agent research — quick, standard, extensive, and deep modes
Scraping	Web scraping via Bright Data proxy and Apify social media actors
Security	Recon, web app testing, prompt injection testing, security news
Telos	Life OS — goals, beliefs, wisdom, project dashboards, McKinsey reports
Thinking	First principles, council debates, red team, brainstorming, science
USMetrics	68 US economic indicators from FRED, EIA, Treasury, BLS, Census
Utilities	CLI generation, skill scaffolding, Fabric patterns, Cloudflare, browser automation

Each pack works standalone — install one, install five, or install all of them. They're designed to give you PAI-level capabilities whether or not you run the full PAI system.

Browse all packs →

❓ FAQ

How is PAI different from just using Claude Code?

PAI is built natively on Claude Code and designed to stay that way. We chose Claude Code because its hook system, context management, and agentic architecture are the best foundation available for personal AI infrastructure.

PAI isn't a replacement for Claude Code — it's the layer on top that makes Claude Code *yours*:

- **Persistent memory** — Your DA remembers past sessions, decisions, and learnings
- **Custom skills** — Specialized capabilities for the things you do most
- **Your context** — Goals, contacts, preferences—all available without re-explaining
- **Intelligent routing** — Say "research this" and the right workflow triggers automatically
- **Self-improvement** — The system modifies itself based on what it learns

Think of it this way: Claude Code is the engine. PAI is everything else that makes it *your* car.

What's the difference between PAI and Claude Code's built-in features?

Claude Code provides powerful primitives — hooks, slash commands, MCP servers, context files. These are individual building blocks.

PAI is the complete system built on those primitives. It connects everything together: your goals inform your skills, your skills generate memory your memory improves future responses. PAI turns Claude Code's building blocks into a coherent personal AI platform.

Is PAI only for Claude Code?

PAI is Claude Code native. We believe Claude Code's hook system, context management, and agentic capabilities make it the best platform for personal AI infrastructure, and PAI is designed to take full advantage of those features.

That said, PAI's concepts (skills, memory, algorithms) are universal, and the code is TypeScript and Bash — so community members are welcome to adapt it for other platforms.

How is this different from fabric?

Fabric is a collection of AI prompts (patterns) for specific tasks. It's focused on *what to ask AI*.

PAI is infrastructure for *how your DA operates*—memory, skills, routing, context, self-improvement. They're complementary. Many PAI users integrate Fabric patterns into their skills.

What if I break something?

Recovery is straightforward:

- **Back up first** — Before any upgrade: `cp -r ~/.claude ~/.claude-backup-$(date +%Y%m%d)`
- **USER/ is safe** — Your customizations in `USER/` are never touched by the installer or upgrades
- **Settings merge, not overwrite** — The installer only updates identity and version fields; your hooks, statusline, and custom config are preserved
- **Git-backed** — Version control everything, roll back when needed
- **History is preserved** — Your DA's memory survives mistakes
- **DA can fix it** — Your DA helped build it, it can help repair it
- **Re-install** — Run the installer again; it detects existing installations and merges intelligently

---

🎯 Roadmap

Feature	Description
Local Model Support	Run PAI with local models (Ollama, llama.cpp) for privacy and cost control
Granular Model Routing	Route different tasks to different models based on complexity
Remote Access	Access your PAI from anywhere—mobile, web, other devices
Outbound Phone Calling	Voice capabilities for outbound calls
External Notifications	Robust notification system for Email, Discord, Telegram, Slack

---

🌐 Community

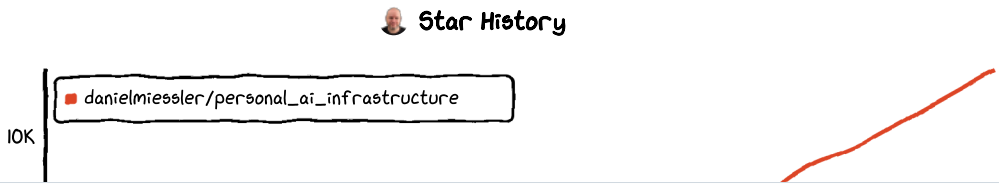
GitHub Discussions: Join the conversation

Community Discord: PAI is discussed in the community Discord along with other AI projects

Twitter/X: @danielmiessler

Blog: danielmiessler.com

Star History



Releases 19

PAI v4.0.3 — Community PR Patch Latest
on Mar 2

+ 18 releases

Sponsor this project

Packages

No packages published

Contributors 25

+ 11 contributors

Languages

TypeScript 76.3% **Python** 9.8% **Shell** 4.6% **Vue** 3.1% **JavaScript** 2.6% **CSS** 1.8% **Other** 1.8%