Workforce OS

A file-based multi-agent operating system for a 7-person team.

```
$ no database. no server. just markdown and git.
```

GAURAV KEERTHI · CEO, STRONGKEEP
~30 MIN · TECH SHARE

The mornings started the same way.

- Seven people. Four hats each. **Everyone** context-switching.
- `"What was I working on yesterday?"`
- `"What did Gaurav promise that customer?"`
- `"Did anyone follow up on the partner deck?"`
- Every AI session a cold start. Ten minutes briefing the assistant — exactly the ten I was trying to save.

→ Multiply by seven people × every day. The overhead *is* the product.

Tourists.
Not residents.

- Copilot, ChatGPT — powerful, `stateless`. Memory dies with the session.
- They live in a sidebar. My work lives in M365, Atlassian, the terminal.
- Per-user assistants only. Nothing shared across the team.
- Nothing enforces scope. `"Don't send email"` is a prayer, not a rule.

→ The assistant has to be a *resident* of the workflow. Not a visitor.

Use the system. Don't fiddle with it.

- Pick one ecosystem. Commit. **Claude** — Code, mobile, connectors. A walled garden that works.
- AI `buddies` to supercharge each teammate. AI `staff` for work nobody's doing.
- Coordination needs structure — rules, boundaries, handoffs — not vibes.
- Plug into the existing stack (M365, Atlassian, HubSpot). Don't make the team change tools.
- Built for 7 people. Not 7,000.

`? The rest of this talk is what fell out of trying to answer that`

Six principles.
All non-negotiable.

- **Files are the system.** · `ls` is the dashboard. `git log` is the audit trail.
- **Conventions over configuration.** · One template. One shape. Zero settings panel.
- **Security by design.** · Agent's identity, access, and mission/mandates are all in separate architectural layers.
- **Minimise hidden state.** · If it's not in a committed file, it doesn't exist.
- **Visible failure.** · Scripts fail loudly. No silent fallbacks.
- **Non-destructive by default.** · Agents draft. Humans approve.

09:45 SGT.
My brief, before I ask for it.

KNIGHT/WORKSPACE/BRIEFS/2026-04-22.MD

```
# Morning Brief — 2026-04-22 (Wed)

Today:
- 10:15 Retreat disc. w/ Zaishao
- 13:35 Black Hat panel, MBS L4
- 15:00 Informa interview

Top 3 priorities:
1. Black Hat panel — execution day
2. MTX deck build — Thu PM + Fri AM
3. Retreat prep before 10:15 sync

Carryovers:  SKA-3 (overdue)
Chase:       SKA-11, SKA-12 OKRs
Flag:        Guard 11:30-13:00
```

WHAT KNIGHT PULLED TO WRITE THIS

- today's M365 calendar
- tomorrow's calendar (for prep)
- Jira — SKA + STK + SKP + SR
- Outlook To Do
- yesterday's brief, for carryovers
- meeting transcripts, for verbal commitments

→ Lands 15 min before the 10:00 standup.

→ Next session, Knight already knows. No cold start.

Five folders.
Every agent, same shape.

THE SYSTEM

```
workforce-os/
├── platform/        # rules, schemas
├── agents/          # per-agent
├── shared/          # cross-agent
├── integrations/    # M365, Atlassian MCPs
└── ops/             # bash scripts
```

→ At launch, these concat into a single prompt.

EACH AGENT

- `soul.md` · personality, voice
- `profile.md` · role, boundaries
- `memory/` · context, routines, lessons
- `tasks/` · inbox → active → done
- `workspace/` · private scratch

Each human gets one named agent.

- **Knight** assists Gaurav · **Harold** assists Clement · **Carlton** assists Zaishao…
- The agent is **not** the human. It is their assistant.
- Profile boundaries: `never sets strategy · never approves · never sends.`

SIR STONK · CHIEF OF STAFF

No human pair. Triages tasks. Routes to owners. The only entity that can promote drafts to shared knowledge. Prevents agent-to-agent chaos.

One command. That's the interface.

```
$ workforce          # launch your agent
```

At launch: identity checked → MCP config generated → prompt assembled (base + soul + profile + memory + active tasks) → Claude Code session starts with a named /remote-control session ("Knight 230426") so users can opt for the Claude App GUI instead of Terminal. Regular health checks, tasking, and syncs happen hourly.

→ The agent picks up exactly where you left off.

The filesystem is the Kanban board.

```
T-20260421-0003.md

    inbox/  →  active/  →  done/
                  ↑
            (you are here)
```

- Location = status. No separate state field to drift.
- Filename has date + atomic counter. No collisions.
- `ls tasks/active/` shows your workload. `git log` shows history.
- Cross-team? Link to a Jira issue via `jira_key:` in frontmatter.

Black Hat Asia, in 48 hours.

One task. Three systems. Traced end-to-end.

```
Sun 20 Apr 10:20  T-20260420-0001 filed in knight/tasks/active/
                  · Sonnet subagent researches co-panellists (~90s)
                  · Panel prep v1 drafted

Sun 20 Apr 10:25  Published → Confluence WOS > Knight (pg 253362197)

Sun 20 Apr 10:30  v2 — 5 calibrations from Gaurav applied

Wed 22 Apr 13:35  Panel delivered.  Task → done/

Wed 22 Apr 18:55  Signed commit pushed.  Audit trail complete.
```

THREE SYSTEMS, ONE THREAD

- `knight/tasks/` · local filesystem
- `WOS > Knight` · Confluence
- `git commit` · signed, auditable

WHAT I DIDN'T DO

- Re-explain the panel to Knight.
- Chase prep across three apps.
- Lose state when the session ended.
- Forget a calibration I gave on Sunday.

Calendar → LinkedIn post. One thread. Six systems.

A media interview, start to finish. No hop was manual.

```
Mon          Knight spots Informa interview on my calendar      [M365]
             pulls the email thread with Naomi's questions      [M365]
             web-searches Naomi + Informa's recent coverage     [web]
             drafts a one-pager brief in knight/workspace/      [fs]

Wed 15:00    I take the call. Fireflies records.                [Fireflies]

Wed 16:00    Transcript drops. Carlton pulls it.                [MCP]
             summary + soundbites → shared/knowledge/meetings/  [fs]

Wed 18:00    Chloe drafts a LinkedIn post from the soundbites   [autonomous]
             files an SKA ticket for my approval                [Jira]
             I review, edit, approve → post ships               [human]
```

→ Six systems. One workflow. I showed up for the interview. The rest was orchestration.

Chloe runs hourly. With no tools. At all.

Chloe · BizDev & Marketing · hourly cron · no human at runtime. Her Claude instance has *no filesystem, no bash, no MCP*. A Python driver passes context in; Claude returns structured output; the driver writes files and files tickets.

THE CYCLE

```
kill switch
  ↓ concurrency lock
  ↓ circuit breaker
  ↓ budget cap
  ↓ pre-workers (seed context)
  ↓ claude → JSON only
  ↓ post-workers (file Jira)
  ↓ signed commit + push
```

If a layer trips, the cycle is meant to abort — loudly.

WHAT SHE WROTE THIS WEEK

```
id:    T-20260418-1000
kind:  pending-action
verb:  publish-linkedin
title: "Data residency
         trends in ASEAN"
hash:  dd520e90…b3d
```

→ driver files it as SKA-3 · assigned: Gaurav · status: To Do

→ Every draft sits behind my approval. I review, edit, or kill it.

Twelve layers.
So no one layer has to be perfect.

- 01 Zero tools
- 02 Path-scoped commits
- 03 Jira-gated approval
- 04 Per-agent secrets · 600
- 05 Egress allowlist
- 06 Prompt-injection defences
- 07 Signed commits · SSH
- 08 Circuit breaker
- 09 Kill switch
- 10 Concurrency lock · flock
- 11 Budget cap · daily USD
- 12 Gitleaks secret scan

→ Defence in depth. Each layer reduces risk; none is a guarantee.

What we gave up.

- **Real-time collaboration.** Git push/pull is the latency.
- **Fancy UI.** It's terminals and markdown. If you like clicking, look elsewhere.
- **Horizontal scale.** Git strains around 15–20 agents. Not a 1,000-agent system.
- **Always-on for every role.** Paired agents launch on demand. Chloe & Sir Stonk are persistent — more autonomous roles (sales, HR) to follow.

→ Not a panacea. Honest about the ceiling.

What we gained.

- **Perfect auditability.** Every change is a signed commit.
- **Zero infrastructure cost.** Laptops + a free git remote.
- **Debuggable by reading.** No black boxes. Read the files. Read the prompt.
- **Model-portable.** Swap Claude for another model — it's just a prompt.
- **Nothing to breach.** No database. No server.

Four things I didn't expect.

1. Personality files (`soul.md`) reduce drift. Voice is cheap to give — give it.
2. Explicit boundaries catch most scope mistakes before they happen. Most agent failures are scope failures.
3. File-based memory that survives sessions is a *superpower* .
4. *"Draft only, never send"* is a sensible Phase 0 for any autonomous agent.

Questions I'm still grappling with.

1. Does the single-orchestrator (CoS) pattern scale — or does it become a bottleneck at 15+ agents?
2. Is file-based memory scalable? Append-only files grow forever; so does the prompt.
3. Security vs. integrations: how do we contain blast radius as we add tools and agents?
4. How do autonomous agents earn trust to graduate from *draft only* to *send*?
5. How do we measure leverage — are we shipping more, or just looking busy?

? If you've solved any of these, I'd love to hear it

THANK YOU. · DISCUSSION.