David (wizzense)

Outcome Engineer. Founder & Architect, Aitherium / AitherOS. USAF veteran.

remote · david.parkhurst@aitherium.com · github.com/wizzense · github.com/Aitherium · linkedin (on request) · blog.aitherium.com

I run a software factory, not a backlog. Over 14 months I built AitherOS — an AI-native operating platform where 33 specialised agents plan, code, review, audit, deploy and write about their own work — and used it to land 21,065 commits, 642 self-testing quality gates, ~1,200 MCP tools and 270 published posts, most of it authored and shipped by agents under programmatic governance I designed. Before that: 12 years racking, securing and automating enterprise infrastructure, including the second-largest Active Directory on earth for the US Air Force. I have scars from every layer of the stack and a documented, measured opinion about how autonomous systems fail. Outcome engineering — orchestration, governance, memory, evaluation, self-healing, ship — is the job I have been doing, in the open, every day.

21,065 commits since Jun 2025	**13,928** commits, last 90 days	**642** self-testing quality gates	**~1,200** MCP tools served	**369** services declared	**270** posts published by agents

OUTCOME ENGINEERING, THE CHARTER, POINT BY POINT

ARCHITECT MULTI-AGENT SYSTEMS

Designed the AitherOS dispatch ladder: a 33-agent roster (Demiurge codes, Hydra reviews, Athena audits, Apollo profiles, Prometheus deploys, Scribe documents) behind a single MCP gateway, with capability-based routing, swarm coding (11 parallel LLM agents), agent-to-agent mesh messaging (awrelay), and a ratchet loop (propose -> apply -> score -> keep/revert) driving measurable improvement. All LLM traffic routes through one priority scheduler (MicroScheduler) across a local RTX 5090, a DGX Spark pool and rented cloud GPUs.

ENGINEER CONTEXT AND MEMORY

Built the semantic substrate agents work from: a code graph (awgraph, AST + tree-sitter call graphs, recall@25 measured), a repo-understanding layer (repowise), a portable scoped memory (awm), a knowledge graph with hybrid search (Nexus), a 25-question retrieval gold set with canaries that gate every chunking/embedding change, and a per-project doctrine file bind-mounted into every agent workspace. Trained a code-search embedder 13x smaller than its teacher that beats it, for $3 of GPU.

DESIGN SELF-HEALING LOOPS

Autoheal + a death watchdog (opt-out) restart wedged services; a 4-hourly probe runs 79 live gates in-fleet and pages on VIOLATION and on DEAD separately; a data-writability gate auto-repairs non-sensitive permission drift and quarantines anything credential-shaped; a paging-path gate asserts the page itself LANDS by round-trip, after an outage where detection worked and the page died. Root-caused and documented the incidents (nine-hour WSL flap,

IMPLEMENT AUTOMATED GOVERNANCE

Replaced manual review with 642 checker gates, each required to carry a --self-test proving it can still fail, each wired into one of four unattended lanes (CI, 4-hour in-fleet probe, host probe, self-test-only with a stated reason). A ledger rule forbids recording a mechanically detectable defect as a note: it must become a check. Stop hooks block a session that changed code without touching a gate or the ledger. A written "no false blockers" doctrine defines exactly three legitimate reasons an agent may stop (physically impossible, irreversible and destructive, owner said no) and measurably cut stalled loops.

BUILD RIGOROUS EVALUATION FRAMEWORKS

Every gate exits 0 / 1 / 2 (pass / violation / could-not-judge) and "could not judge" is never a pass. Built the gauntlet: capture -> change -> recapture -> blind A/B verdict -> raw-exit-code gate -> accept/reject with a rejected ledger. Caught a trainer that reported "3 epochs over 20,000 records" while touching 192 rows, an optimiser that had never run a round while logging "completed", and a mock-driven test suite that never executed the loop it tested. Bucketed metrics by seen/unseen so an aggregate at ceiling cannot hide a bucket at floor.

PROTOTYPE AND DEPLOY

Ship cadence: a new tenant app (backend, agent brain pack, portal, Cloudflare tunnel, GitHub Pages) in about ten minutes; three paying or pilot tenant appliances live (repair shop, game studio, political campaign tooling); 61 public repos; four packages on PyPI; an on-device 1.7B model running in a Pixel 10 browser talking to the platform; a WebMCP design studio in the browser; and every blog post converted to a narrated video by the pipeline that made the film on wizzense.github.io.

thundering-herd boot storm, silent-empty SSE lanes) as gates, not memories.

EXPERIENCE

Founder, Architect (AitherOS, the aw* stack, awnix) — Aitherium — Jul 2025 - Present · Remote

- Architected AitherOS: 369 declared services across 12 layers, run as podman quadlets on Rocky/Debian, with Postgres replication, Redis Sentinel, Cloudflare tunnels, and GPU scheduling across local, DGX and rented (Lambda, Vast, Hetzner, AWS) capacity.
- Built the aw* (Aither World) family: 64 registered bricks, 40 public, composed into 14 stacks. awnix is an immutable Linux you can hand to an agent; awdk builds agent fleets in three lines; awgit is op-log-aware git for many agents in one tree; awgraph, awm, awrelay, awiam, awbac, awseal, awdit cover code graph, memory, messaging, identity, RBAC, signing and audit. Four on PyPI.
- Governance as code: 642 self-testing gates, a per-gate lane manifest (CHK001-006) that asserts 100% coverage and refuses a gate that runs nowhere, and a debt-ledger protocol where a detectable defect must become a check. Measured the ledger at 79 rows/day in vs 4.2/day out before the rule; it stopped being a write-only archive after.
- 21,065 commits since June 2025 (13,928 in the last 90 days), 33,075 tracked files, 270 blog posts, 160 scheduled routines - the majority authored by agents under those gates, with every claim required to carry its command and exit code.
- Trained and shipped models: a code-search embedder 13x smaller than its teacher that beats it (recall measured on the platform's own corpus); Gemma-4 and orchestrator fine-tunes with a positive-baseline rule (a trainer must beat a named baseline or it is not a result); Bonsai 1.7B running in-browser on a phone.
- Security posture designed fail-closed: authz gates deny on every non-happy path, no caller-supplied tenant ids in a decision, dual-key rotation of the fleet credential across 43 verifier sites, secrets vault with audit round-trip asserted by a gate.
- Sold and operated real tenants: J Games & Repair, Digital Ground Game, Garg - each a generated repo, appliance image, brain pack and portal the customer owns.

Senior Enterprise Services Engineer (prev. Escalation Engineer, Senior TAM, ESE) — Tanium — Jan 2022 - Present · Remote

- Platform SME on the hardest escalations for enterprise and federal endpoint-management customers; 195 cases in six months at 100% CSAT.
- Rewrote the Tanium Risk Assessment tooling in Python to scale from 500 to unbounded endpoints; applied to a 50,000-endpoint environment.
- Built Client Health dashboards and automation moving customers toward a self-healing endpoint estate; authored the guide and ran the live webinar (hundreds of attendees).
- Fully automated a home lab through Tanium + PowerShell + Python as a self-managing environment - the seed of AitherOS.
- 2x Star Performer award (2024).

HPC Linux System Engineer — Boeing — Apr 2020 - Jan 2022 · Seal Beach, CA

- Owned the enterprise tape archive (Oracle HSM / SAM-QFS -> Versity Storage Manager as technical lead) and a fibre-channel SAN for HPC.
- Wrote Python automation for recovery, monitoring and archival; led LTO6 -> LTO8 migration and data-at-rest encryption (OKM 3, IBM SKLM).

Enterprise Active Directory Engineer, Functional Lead, Supervisor Directory Services Ops — United States Air Force - Cyber Systems Operations — Jan 2014 - Jan 2020 · Honolulu HI; Lajes Field, Portugal; Keesler AFB

- Led Tier-2 operations for a $10B, 850,000-user, 690-domain-controller, 230-site Active Directory - the second-largest AD network in the world - including classified and unclassified GAL/Exchange services 24/7.
- Automated DC maintenance and health checks for 70+ domain controllers in PowerShell; performed an authoritative ADDS restore recovering 3,000 accounts with zero collateral impact.
- Identified a training gap, wrote the ADDS troubleshooting curriculum, and trained 357 technicians across four organisations.
- Red Flag 17-1 blue-team operator, Superior Performer; Junior Technician of the Year 2017; Information Dominance award 2018; promoted SSgt in the 94th percentile.

SKILLS

Agentic systems: multi-agent orchestration, MCP (1, 200+ tools served), tool-calling harnesses, agent memory and knowledge graphs, evaluation harnesses, ratchet/keep-revert loops, prompt and context engineering, on-device inference (WebGPU/wasm)

Models and training: vLLM, llama.cpp, LoRA/fine-tuning, embedding distillation, KV-cache quantisation, GPU scheduling, DeepSeek/Qwen/Gemma/Bonsai

Languages: Python, TypeScript/Node, PowerShell, Bash, SQL

Platform: PostgreSQL (replication), Redis Sentinel, Podman/Docker, systemd quadlets, k3s, Next.js/React, FastAPI, Cloudflare (Workers, Tunnels, Pages), GitHub Actions self-hosted fleets, OpenTofu

Cloud and hardware: AWS, Lambda, Vast, Hetzner, NVIDIA DGX Spark, RTX 5090, WSL2

Ops discipline: fail-closed authz, secret rotation, observability-to-action loops, incident root-cause, tape/SAN/HPC storage, Windows and Linux at 850K-user scale

CERTIFICATIONS

Tanium Certified Professional - Endpoint Management (2024-2026) · Tanium Certified Operator (2022-2026) · GIAC Certified Windows Security Administrator (GCWN) · GIAC Certified Enterprise Defender (GCED) · CompTIA Security+

PROOF

AitherOS / aw* org: github.com/Aitherium · Blog (270 posts, agent-written and gated): blog.aitherium.com · 13x smaller embedder beats its teacher: blog.aitherium.com/code-search-embedder-13x-smaller-beats-its-teacher · Live demo hub: demo.aitherium.com · Who I am (this site): wizzense.github.io

Every figure above was measured on 2026-09-03; the commands are in facts/measured.md in the repository behind wizzense.github.io. This document was assembled by the AitherOS agent platform and reviewed by its author.