TYLER YOUK

(719) 205-9983 | dev@tyleryouk.com | linkedin.com/in/tyler-youk | github.com/kxagent | tyleryouk.com

EDUCATION

George Mason University, *Bachelor of Science (BS), Computer Science, GPA: 3.74*

PROFESSIONAL EXPERIENCE

Kratos Defense & Security **June 2025 - Present**

Software Engineer

Ground-Station Observability Platform

- Rebuilt and own a Python Prometheus exporter that turns four classes of ground-equipment health (databases, service APIs, host diagnostics, AWS EC2) into 27 metric families, at 99.97% scrape availability in production.
- Designed and built a background-polling data-flow exporter end to end (Python, Helm chart, Grafana dashboards, 13 alert rules), cutting scrape cost from 55 seconds to 40 milliseconds.
- Eliminated false health signals across distributed ground-equipment endpoints: fixed a roll-up stuck at 0% healthy, and rebuilt a database check that read green through real outages on a direct connection-status gauge.
- Built, as sole engineer, the alerting application a 24/7 operations center uses: Angular, FastAPI, WebSockets and PostgreSQL, with database-first override writes, versioned real-time updates, Keycloak SSO and automated tests in CI.
- Moved dashboards and alert rules to GitOps with Helm and Grafana sidecars on a Prometheus stack of about 440,000 active series at 6,300 samples per second, and migrated services from Docker Compose to Kubernetes.

Satellite Collection-Tasking Platform

- Extended a FastAPI and PostgreSQL tasking API so one endpoint schedules both geostationary and low-orbit satellite observations, adding six REST endpoints and a per-antenna work queue without breaking existing callers.
- Authored the geostationary tasking specification and a generated OpenAPI contract of 19 operations, then built the implementation across the API, schema, collection agent and web UI.
- Designed four tasking-schema tables with integrity constraints, and caught before merge a trigger and column-type mismatch that would have failed every insert.
- Moved the Angular tasking UI off a mock backend onto live inventory, catalog and SSO services, with role-based access and automated UI tests in CI.

SKILLS

Languages & Backend: Python, SQL, TypeScript, Bash, FastAPI, REST and WebSocket APIs, PostgreSQL, SQLAlchemy

Observability & Reliability: Prometheus, PromQL, Grafana, Alertmanager, OpenTelemetry, monitoring, alerting, SLOs

Cloud & Infrastructure: AWS GovCloud, GCP, Kubernetes, Helm, Docker, Linux, systemd, Tailscale, Cloudflare Tunnel

IaC & CI/CD: OpenTofu (Terraform-compatible), Ansible, Molecule, GitHub Actions on self-hosted runners

Security & Secrets: Infisical (self-hosted), HashiCorp Vault, SOPS + age, Keycloak, OAuth 2.0, gitleaks

Agentic Engineering: Claude Code (hooks, skills, subagents), OpenCode, T3 Code, Claude Agent SDK, FastMCP, A2A, RAG, parallel subagent fan-out, subagent context isolation, quota-aware scheduling, LLM-judge evals

PROJECTS

Self-Hosted Agent Platform: Infrastructure as Code on One Linux Server

- Built a self-hosted agent orchestration platform: board-driven dispatch across worker lanes with quota-aware scheduling and attempt caps, each run a transient systemd unit in its own git worktree under a concurrency lock.
- Provisioned the on-prem server as code with configuration management: 10 idempotent Ansible roles converged hourly by ansible-pull with Molecule tests, and OpenTofu for Cloudflare, Tailscale and 36 GitHub repositories.
- Hardened the host to the CIS Level 1 benchmark (68.47 to 87.60) one section at a time, with PAM and sshd changes behind a 15-minute auto-restore timer and no failed units or agent downtime.

Atlas: Org-to-Org Agent Gateway

- Built an always-on A2A v1.0 gateway with an HTTP+JSON API, SSE streaming and push webhooks, a published Agent Card, and per-peer Bearer tokens stored as SHA-256 hashes and compared in constant time.
- Built a durable SQLite inbox with at-least-once delivery and idempotent, store-before-ack processing: messages are deduplicated on message ID, drained in order and retried without double-sending.
- Specified an inter-org agent protocol on A2A with five typed message types, identity tied to each company's domain, cross-org trust levels and human escalation, proposed to a partner company.