

# Kova: On-Device AI/ML for Autonomous Modular Mission Payload Development

## The Cochran Block, LLC

ARCYBER · AI Effects at the Edge Assessment Event

Whitepaper Submission · April 2026

Principal Investigator: Michael Cochran · Army 17C, JCAC 2014, USCYBERCOM J38 JMOC-E

CAGE 1CQ66 · UEI W7X3HAQL9CF9 · SAM.gov Active · SDVOSB Pending

Contact: mcochran@cochranblock.org · 443-900-7903 · cochranblock.org

## 1. Executive Summary

The Cochran Block proposes to extend **Kova** — an existing, working on-device AI augment engine — into a modular mission payload (MMP) development platform for ARCYBER. Kova is a single-binary Rust application that runs local large language model inference, executes a 13-tool agent loop, and orchestrates work across distributed worker nodes via SSH. It does not call external APIs, does not require cloud connectivity, and does not exfiltrate data. Today it is published as `kova-engine` on crates.io and runs in production for our internal development workflows.

For this assessment event, we propose using Kova as the AI/ML platform that ARCYBER seeks: an embedded model that pulls from supported data repositories, autonomously develops low-sophistication MMPs, and enables porting payloads to connected sensors. The architecture is compiled, container-friendly via Infrastructure-as-Code, and built on memory-safe Rust — aligning with ARCYBER's DevSecOps standards and providing a clear path to ATO through reduced attack surface.

**Honest TRL assessment:** Kova as a general agentic AI platform is approximately TRL 5 — component validated in a relevant environment (production use on the proposer's own development workflows, 314 tests, 96 source files). The MMP-specific functionality, sensor integration, and ATO package do not yet exist; this whitepaper proposes building them under Phase 1. We are not claiming a deployed mission system. We are proposing to extend a validated component into a mission-relevant prototype.

## 2. Alignment to Requirements

| ARCYBER Requirement                         | Existing Kova Capability                                                | Phase 1 Build                                                                      |
|---------------------------------------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| AI/ML model embedded on commercial platform | Yes — Kova embeds GGUF models via Kalosm; runs entirely local           | Add ARCYBER-supplied model artifacts; integrate with mission-specific data schemas |
| Data repository integration (pull/push)     | Partial — Kova reads/writes local sled database, calls APIs via request | Add ARCYBER repository connectors; standardize data normalization                  |

|                                      |                                                                                                                                    |                                                                         |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Autonomous MMP development           | Component — agent tool loop autonomously generates and validates code via 13 tools                                                 | Define MMP grammar; train classifier on existing payload examples       |
| Sensor integration / payload porting | Not implemented                                                                                                                    | New — build sensor adapter layer using existing C2 mesh architecture    |
| Containerized Infrastructure-as-Code | Partial — single-binary deploy is reproducible; not yet packaged in OCI                                                            | Add OCI container manifest, Helm chart, Terraform module                |
| DevSecOps and ATO compliance         | Component — memory-safe Rust eliminates buffer overflows, use-after-free, data races; embedded SBOM and SSDF docs in every release | Build full ATO package: STIG mapping, CCI traceability, eMASS artifacts |

### 3. Technical Approach

---

#### 3.1 Core Architecture

Kova compiles into a single Rust binary (currently 30 MB with full features) that contains: a local LLM inference engine, an agentic tool loop, an embedded sled key-value store, an HTTP interface, and a distributed orchestration layer. There are no external runtime dependencies beyond the operating system. Deployment is a single file copy.

For this effort, the AI/ML model is loaded as a GGUF artifact at startup, signed with NanoSign (a 36-byte BLAKE3 integrity signature we developed and use across our portfolio). Models are verified before any inference occurs. Tampered or unsigned models are rejected. This eliminates the supply chain attack surface that container registries and certificate authorities introduce.

#### 3.2 Modular Mission Payload Development

The agentic tool loop (currently 13 tools: read, write, edit, bash, glob, grep, search, code review, and others) provides the autonomous workflow primitive. For MMP development, we propose adding domain-specific tools: *payload\_template\_select*, *parameter\_inject*, *sensor\_capability\_query*, and *payload\_validate*. The loop selects a template, injects mission parameters, queries available sensors, generates the payload, and validates it against ARCYBER's existing security checks.

We are explicit about scope: this approach handles low-sophistication MMPs as defined in the assessment release. High-sophistication payloads require human-in-the-loop review and are not in scope for the Phase 1 prototype.

#### 3.3 Data Repository Integration

Kova currently integrates with eight federal APIs in our existing whobelooking subsystem (SAM.gov, USASpending, Federal Register, Grants.gov, Regulations.gov, CALC, SAM.gov Contract Awards, SBIR.gov). Each integration is a small Rust module with rate-limit awareness, retry logic, and sled-cached results. We will replicate this pattern for ARCYBER's data repositories with the connectors specified during Phase 1 kickoff.

#### 3.4 Sensor Integration and Connected Operations

Sensor integration is the largest gap between current Kova capabilities and the ARCYBER requirement. We propose building a sensor adapter layer using our existing C2 Swarm Mesh — the same SSH-based distributed

orchestration we use to coordinate work across our four-node development fleet. The adapter exposes sensors as nodes in the mesh; payloads are deployed via the same path.

### 3.5 DevSecOps and ATO Compliance Path

Our existing compliance posture: every Rust binary embeds an SBOM, SSDF practice mappings, and named federal compliance documentation. Memory-safe Rust eliminates entire classes of CVE. Triple Sims (a deterministic test methodology we use across all six of our production projects) requires every test suite to pass three times identically before merge. For ATO, Phase 1 includes producing the eMASS-ready package: control implementation statements, STIG checklist mappings, and a continuous monitoring baseline.

## 4. Phase 1 Prototype Plan (6 months, prototype OTA)

| Month | Deliverable                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------|
| 1     | Kova baseline deployed in CFIC test environment; ARCYBER model artifacts loaded and verified via NanoSign    |
| 2     | Data repository connectors built for two ARCYBER-specified repositories; sled cache integrated               |
| 3     | MMP template grammar defined; autonomous payload generation working for two low-sophistication payload types |
| 4     | Sensor adapter layer prototype; one sensor type integrated via C2 mesh                                       |
| 5     | Containerized IaC delivery (OCI image, Helm chart, Terraform module); ATO package draft                      |
| 6     | Demonstration environment ready; documentation, architecture diagram, operational runbook delivered          |

**Phase 2 expansion** (post-prototype): broader sensor integration, full ATO submission, multi-payload type support, integration with ARCYBER's existing development pipelines.

## 5. Past Performance and Team

**Principal Investigator:** Michael Cochran is the sole owner and PI. Background: Army 17C (Cyber Operations), JCAC 2014, USCYBERCOM J38 JMOC-E development lead for a Congressional NDAA-directed offensive cyber operations study. 13 years of defense and enterprise software development. 30% service-connected disabled veteran. SDVOSB pending via VetCert.

### Working artifacts (publicly verifiable):

- **kova-engine v0.7.0** on crates.io — the augment engine described in this whitepaper
- **exopack v0.1.0** on crates.io — Triple Sims testing framework used by 6+ projects
- **any-gpu v0.1.0** on crates.io — vendor-agnostic GPU tensor engine (AMD/NVIDIA/Intel/Apple via WGSL)
- **header-writer v0.1.0** on crates.io — Unlicense header injector for source provenance
- **15 open source repositories** at [github.com/cochranblock](https://github.com/cochranblock) with full Timeline of Invention and Proof of Artifacts documentation

- **cochranblock.org** — production single-binary website (9.9 MB Rust binary, \$10/month) outperforming Booz Allen, Leidos, SAIC, and CACI on first paint and CLS in independent benchmarks at [cochranblock.org/speed](https://cochranblock.org/speed)

**What we have not yet done:** We have not previously delivered a contracted mission payload system. We have not held an ATO. We have not integrated with classified data repositories. The PI's prior offensive cyber operations work was as a uniformed military member, not as a contractor. This whitepaper does not represent us as a system integrator with a track record in mission payload development. We are proposing to apply a working AI augment engine to a new domain.

## 6. Differentiators and Why Kova Fits This Need

- **Already on-device.** Most "edge AI" pitches start by promising they will eventually run local. Kova already does. Today.
- **No cloud assumption.** The architecture does not require connectivity, container orchestration, or vendor support. It runs in DIL (denied, intermittent, limited bandwidth) environments by default.
- **Memory-safe Rust.** Eliminates buffer overflows, use-after-free, and data race vulnerabilities — entire classes of CVE that traditional C/C++ MMP tooling exposes.
- **NanoSign integrity.** 36-byte model signatures verified at load. Smaller than a TLS certificate. No certificate authority required.
- **Triple Sims determinism.** Every change passes three identical test runs before merge. Reproducibility is a hard gate, not a goal.
- **Public, auditable, no vendor lock-in.** All code is Unlicense (public domain). The Government can fork, modify, and redistribute without licensing friction.

## 7. Risks and Honest Limitations

| Risk                                                 | Mitigation                                                                                                       |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Sensor integration is unbuilt                        | Phase 1 milestone 4 explicitly delivers prototype adapter; scope limited to one sensor type                      |
| ATO is a multi-month process even with documentation | Phase 1 delivers package draft; full ATO is Phase 2                                                              |
| PI is sole personnel; no team yet                    | Phase 1 funds first hire (cleared developer); meanwhile PI plus subcontracted SME network                        |
| MMP domain knowledge is new to the proposer          | Phase 1 month 1 includes ARCYBER subject matter expert pairing; whitepaper does not claim existing MMP expertise |