Local Intelligence Infrastructure for Multi-UAV Operations

A Technical Note on Fleet Memory Systems
Varun Sharma | November 2025

The Problem:

Indian armed forces are deploying UAVs at unprecedented scale- Herons along the LAC, indigenous platforms like Rustom, and thousands of smaller tactical drones. We're buying good hardware from Israel and the US, building our own, and getting operationally capable fast. But there's a software gap nobody's addressing: these drones don't learn, don't remember, and don't share intelligence.

Here's what happens today:

A Heron drone surveys a 50km stretch of the LAC from 6 AM to 10 AM. It captures video, logs coordinates, maybe flags something unusual. The data goes to a ground station, gets reviewed by an analyst, and filed away.
At 2 PM, another drone covers the same area. It starts from zero. No context about the morning flight. No knowledge of what was seen before. No pattern recognition across flights.
Multiply this by 100 drones, 50 sectors, 365 days. The amount of intelligence being lost is staggering.

Specific operational problems:

1. No mission continuity. Shift A's findings don't inform Shift B's flight plan.
2. Pattern blindness. A vehicle that appears 5 times over 3 weeks across different drone feeds- no system connects the dots.
3. Redundant surveillance. Multiple drones scan the same terrain because there's no shared knowledge of coverage.
4. Learning failure. A drone encounters a difficult target (camouflaged position, terrain shadow)- that experience doesn't help the next drone.
5. Swarm impossibility. You can't coordinate 50 drones if they can't share real-time intelligence.

Why this isn't solved:

Most UAV software is built around video streaming and telemetry. The intelligence layer- memory, pattern recognition, fleet coordination- is minimal or non-existent. The systems that do exist require constant cloud connectivity, which doesn't work in:

- Border areas (Ladakh has dead zones)
- Contested environments (jamming, denial)
- Sensitive operations (can't uplink to foreign servers)

What's Actually Needed

Not a database. Not cloud storage. But a local intelligence layer that works like human memory:

Key requirements:

1. Works offline completely. No cloud dependency, no uplink required. The drone's onboard computer or a ground station edge device runs everything.
2. Fast enough for real-time. Sub-5ms retrieval so mission planning can query "what do we know about Grid 37-N?" and get instant answers during flight.
3. Semantic, not just keyword. If you ask "camouflaged positions near ridgeline," it should find relevant observations even if those exact words weren't used.
4. Shared across fleet. When drones have connectivity (back at base), they sync knowledge. One drone's experience becomes the fleet's knowledge.
5. Pattern recognition. Not just store/retrieve, but identify: "This vehicle has appeared 6 times in different locations" or "This terrain pattern indicates possible concealment."

6. Lightweight. Runs on military-grade edge hardware (Jetson, similar), not server racks.

How This Could Work (Technical Approach)

I've been building something adjacent to this problem- a local AI memory system for edge devices. Here's the core architecture:

Storage layer:

Uses embedded database (RocksDB) with compression- runs on small hardware, handles millions of entries, works offline indefinitely.

Intelligence layer:

Every observation gets converted to a vector embedding (mathematical representation of meaning). This lets you do semantic search- find similar situations even if the words are different.

Retrieval:

HNSW vector index gives sub-5ms query times. Fast enough to query during active missions: "Show me everything relevant to what I'm seeing now."

Sync protocol:

When drones return to base, they sync new memories to the fleet database. Next mission carries forward collective knowledge.

Example use case:

Morning flight logs: "Unusual vehicle movement at 34.2°N, 78.4°E, heading northwest"
Afternoon flight, 10km away, queries: "Any recent activity in this sector?"
System responds in 3ms: "Vehicle matching description logged this morning, projected path suggests possible destination near your current position"

Operator can now make informed decision based on pattern, not isolated data point.

What Makes This Different

Vs commercial UAV software:

- Those need cloud (we're offline-first)
- Those are built for civilian use (we handle military operational tempo)

Vs traditional databases:

- Traditional DB: "Find entries with keyword 'vehicle'"
- This system: "Find situations semantically similar to what I'm seeing" (different retrieval paradigm)

Vs cloud AI:

- Cloud has 200-500ms latency (unusable for real-time)
- Cloud requires connectivity (fails in denied environments)
- Cloud stores sensitive data off-site (sovereignty issue)

Technical Status

I have a working prototype:

- Rust core (performance-critical systems language)
- 2-5ms retrieval
- Python SDK (under development)
- Runs on standard edge hardware
- Fully offline operation

What it's NOT yet:
- Integrated with UAV systems (need military UAV API access)
- Tested with actual mission data formats
- Hardened for field deployment
- Optimized for swarm coordination

What Would Need to Happen

Phase 1- Understand real requirements:
- Work with actual UAV operators to understand data formats
- Study mission planning workflows
- Identify specific queries operators need answered
- Map existing systems we need to integrate with

Phase 2- Defense-specific development:
- Build mission logging adapters (compatible with Indian UAV platforms)
- Add military-specific features (threat classification, terrain analysis)
- Security hardening (air-gapped operation, data isolation)
- Multi-drone sync protocol

Phase 3- Field validation:
- Partner with UAV manufacturer (ideaForge, NewSpace, indigenous platform)
- Test with simulated missions (10+ drones)
- Benchmark performance in realistic conditions
- Get operator feedback, iterate

Timeline: 6-9 months to field-ready system

Why This Matters Strategically

Atmanirbhar angle:

Software brain- the intelligence layer- can be indigenous. Data stays in India, algorithms under our control, no dependency on Israeli or US vendors for the critical thinking layer.

Force multiplier:

100 drones with shared intelligence are more effective than 500 drones operating blind. This isn't about better hardware, it's about making existing hardware smarter.

Future autonomous systems:

This isn't just UAVs. It's the foundation for any autonomous defense system- UGVs, USVs, loitering munitions, swarms. Build the intelligence layer now, apply it across platforms later.

I'd value feedback on whether this direction even makes sense for iDEX consideration, or if there are fundamental gaps in my understanding of the operational environment.

Contact:
Varun Sharma
Phone: +91 9810300618
Based in Gurugram