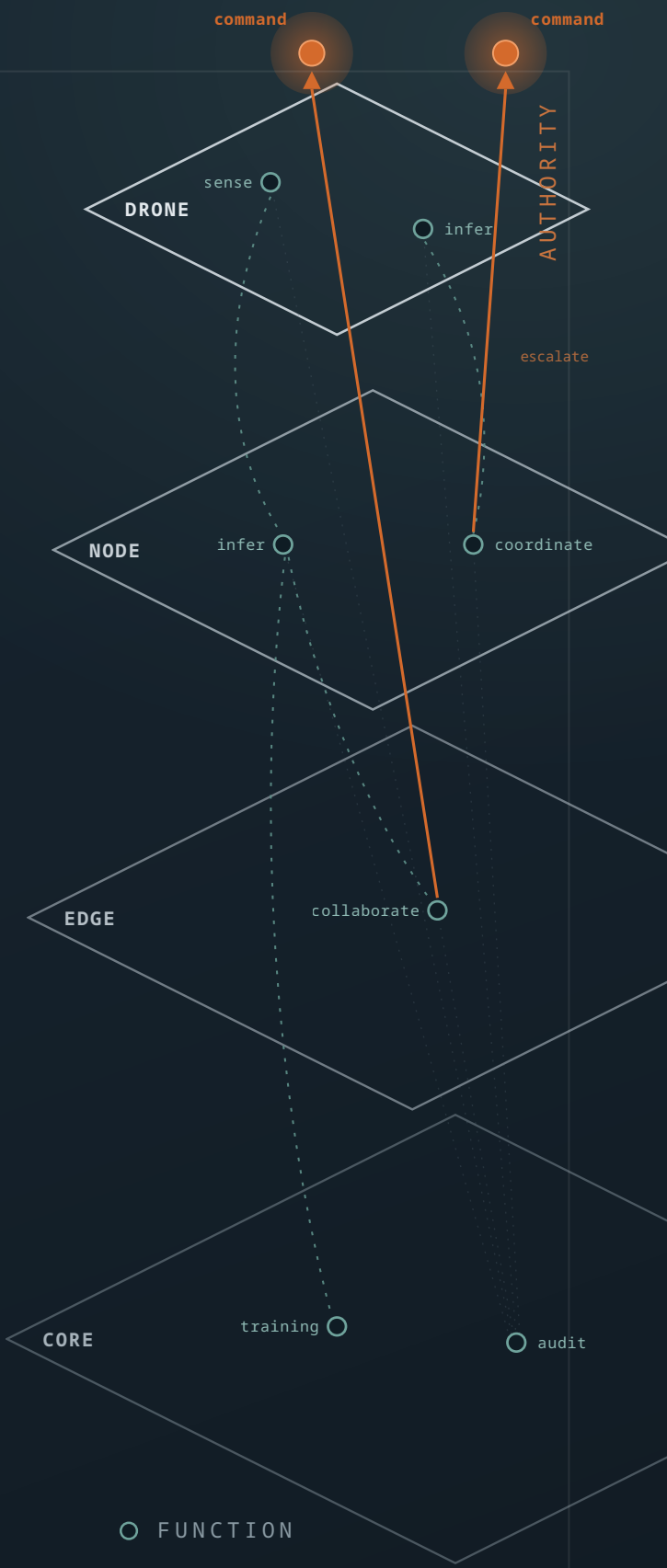
AN APPLIED REASONING ESSAY

Operational AI in Search and Rescue

By **Stephen Alexander**

What a high-urgency mission teaches us about resource contention, function placement, trust under uncertainty, and the cost of being wrong.



One system, decomposed three ways: by **function**, by **placement**, and by **authority**.

EXECUTIVE SUMMARY

This essay tests a simple claim: once AI enters operational missions, it is no longer enough to ask what the model can do. The harder question is how intelligence is decomposed into bounded functions, where those functions run, and what authority their outputs are allowed to carry.

Search and rescue is the forcing function because small changes in connectivity, urgency, governance, and risk make those answers flip. In the first search, a disconnected lone mission compresses the system toward the drone and the mobile command node. The hardest decisions are forced locally: what can be inferred, what can be trusted, and what may be acted on when there is no one upstream to ask.

In the second search, the network holds, but the mission becomes more complex because multiple agencies are involved. The system expands outward. The regional edge becomes more important because the problem is no longer just detection, it is coordination: reconciling what several disparate teams see, deciding what can be shared, preserving provenance, and making low-latency choices without turning every message into a human approval step.

The essay therefore treats architecture as a set of constraint-sensitive decisions rather than a fixed diagram. Each section isolates one decision fork, tests what pulls the system toward the drone, the mobile node, the regional edge, or the core, and identifies the conditions that would make that choice change.

AUTHOR'S NOTE

Why this exists

When I was a kid, my grandfather told me I did not truly know a word unless I could explain its meaning without using the word itself. More than a few times, I struggled through an answer while he waited patiently, sometimes amused, until I found the edges of what I actually understood.

I still think about that when I am trying to learn something new. Do I really understand the idea? Can I apply it? Can I explain it without hiding inside the vocabulary?

This essay tests an idea I have been circling: once AI stops answering single prompts and starts taking multi-step action, holding state, and touching the real world, it becomes a distributed-systems problem rather than a model problem.

Search and rescue is the probe because it stresses that idea without needing a battlefield. It combines urgency, degraded communications, uncertain location, unreliable inputs, privacy concerns, and human accountability. Those constraints force the real questions into view: what functions the system needs, where those functions should run, and what authority their outputs should carry.

These are my own views, not my employer's, and not a forecast from anyone's company. I spend my days around public-sector missions and cloud infrastructure, and some of my own time building AI-enabled applications to see how the pieces fit.

An Applied Reasoning Essay · By Stephen Alexander

Note on method. I used AI tools while developing this essay, primarily to test structure, sharpen phrasing, challenge assumptions, compare alternative framings, and refine the cover, diagrams, and exhibits. I also built and used a personal knowledge base of collected research, notes, and source material to ground the work and check claims as the argument developed. AI was used as a reasoning and production aid, not as an authority. The argument, judgments, final editorial choices, and responsibility for any errors are mine.

CONTENTS

Two searches

Two searches use the same basic equipment, but under opposite conditions. In one, the network fails and the system is forced inward, toward the drone, the local sensors, and the mobile command node. In the other, the network holds, but the mission crosses agencies and jurisdictions, pulling more work toward the regional edge.

An AI model can almost certainly help. The harder question is what kind of system is created once the work is split into bounded functions: sensing, locating, estimating, coordinating, enforcing data rules, preserving provenance, and deciding what may be acted on.

The equipment may look similar in both searches. The architecture does not. As the conditions change, three decisions move with them: what functions exist, where those functions run, and what authority their outputs are allowed to carry.

Missing Hiker

A hiker is overdue, and daylight is running out. A wildfire took out the valley's one cell tower earlier, so there is no signal to the outside world and no way to hand the heavy computing off to a distant data center. Whatever helps find this person tonight has to run on what is physically at the trailhead: several small drones and one command truck. And time is against the search. The temperature is dropping toward the range where an injured person can no longer keep warm, the drones' batteries are draining, and every minute widens the ground the hiker could have reached. The clock does not just tick; it tightens.

The drones go up and search inside a boundary the team drew on the map before launch. Each is doing several things at once in the dark: scanning for a human heat signature, watching its own battery and motors, keeping track of where it is and where on the ground the things it sees actually sit, and judging whether any of it is worth another look.

But thermal search at dusk is not clean. Sun-warmed rock and bare ground can hold nearly as much heat as a body, an animal the size of a person reads much the same from above, and vegetation smears a real signature into the background; the same model that looks sharp in a clean demo is far less sure on a moving, heat-soaked, half-occluded hillside. So across the fleet the drones surface a steady stream of candidates, and a real share of them are wrong or unresolved. Then one comes in that matters more than the rest: a warm shape at the lip of a ravine, about the size of a person, not moving. It could be the hiker. It could be a rock the afternoon sun left warm, or a deer bedded down. Nothing on the drone can settle it alone.

Here is the squeeze. The operator in the truck is not reviewing a clean queue; she is watching feeds, battery timers, map overlays, and radio traffic while a ground team waits

for a coordinate safe enough to act on. If every candidate is pushed to her for a yes or no, she drowns, starts clicking "no" on reflex, and that is the exact moment the real one slips past. But the opposite is worse: you cannot let a machine decide, on its own, that the warm shape is the hiker and send exhausted people down into a dark ravine in an unstable burn scar after what might be a rock.

So the night comes down to a question no one at the trailhead can dodge: with no network to ask and no time to check everything by hand, who is allowed to decide what?

Multi-Agency Search

Now widen the lens. A river has jumped its banks across two counties, the water is still rising, and this time the network mostly still works. The search did not begin as one operation. The county sheriff's team was first on the water and is running the incident; a state rescue team arrived a couple of hours later, a federal task force later still, and each brought its own aircraft, its own people, its own radios, and its own rules about what it may collect, keep, and share.

Their drones even share the same crowded sky and have to be kept clear of one another and of the crewed helicopters working the same water. They are supposed to fold into a single command, and they will, but knitting three organizations into one picture takes hours, and the water is not waiting. On one rooftop a family sits with the current climbing the wall beneath them, and the crews have a narrow window before dark and before the channel shifts.

The trouble is not that no one is in charge; someone is. It is that in the first hours the agencies are not yet working from the same picture. Their search areas overlap, so two crews catch the same rooftop from different angles and log it as two separate rescues; the state team spends a pass on a stretch the county team already cleared, because it did not yet know the county team was there; a federal drone flags a face that seems to match a missing-persons alert, though the record that could confirm or reject the match sits in another agency's system, under rules that do not disappear because the river is rising. And a wrong call does not stay local: a rescue pushed to everyone at once sends boats racing to the wrong roof, and there are not enough boats to waste one.

No single command post can resolve this on its own. Each agency mostly sees its own aircraft, each is bound by its own rules about who may see what, and pulling every feed back to a distant data center to be reconciled is too slow for a rising river; the sensitive parts, a face checked against a protected record, video sweeping across people's homes, may not even be allowed to leave the region. And the calls that matter, sending another agency's boat to that roof or declaring that these are the people who were reported missing, are not ones anyone will let software make.

Something has to operate across all three agencies, reconcile their overlapping picture into one fast, and hand the right piece to the right agency under that agency's rules, without a person approving every message, and be trusted enough that no one commits a boat on a bad tip.

The answer cannot live only on the drone, because the problem is larger than one aircraft. It cannot live only in one agency's truck, because the picture crosses jurisdictions. And it cannot live only in a distant core, because the river, the rules, and the latency budget are all local. The flood raises a different question from the valley: what has to live at the regional edge, and why would anyone believe what it says?

Bounded Functions

The useful unit of analysis is not the model. It is the bounded function. Each scene depended on a portfolio of them: perception, health monitoring, trust assessment, triage, coordination, governance, and planning. Each function had a different tolerance for error, latency budget, compute budget, and failure cost. Each also needed a home.

That is what makes operational AI a distributed-systems problem. The value is in how the portfolio is decomposed, placed, trusted, governed, and allowed to act.

The placement terms in this essay map directly to the two narratives, and they run from the most constrained home to the broadest: the drone, closest to the sensor and fastest to act but bounded by power, compute, memory, and onboard context; the mobile node, the local command post that fuses several nearby drones and keeps one crew working when disconnected; the regional edge, the federation layer that reconciles multiple teams, enforces data rules, and preserves provenance while still acting within the mission's latency budget; and the core, the centralized cloud for training, simulation, large-scale analysis, evaluation, and long-cycle learning. Section 1 defines each precisely; what matters here is that each is a different home with a different budget, and the right home for a function changes with the mission.

The regional edge is also where caching starts to matter at mission scale. Below it, caching is mostly local: recent frames, detections, tracks, battery state, and one crew's working map. At the regional edge it becomes shared and governed: incident summaries, search-sector status, cleared areas, agency rules, policy decisions, provenance records, embeddings, and reusable prompt context. The larger point is that reused context becomes part of the operational substrate.

Those four homes define the placement axis. A function can be assigned to any of them, but the right answer changes with the mission. The disconnected valley pushes work downward. The multi-agency flood pulls work outward.

FUNCTION	LIKELY HOME	MOVEMENT LOGIC
Perception / sensing	Endpoint / asset	Stays closest to the sensor when latency, bandwidth, or disconnection matter. Moves upward only when local compute is insufficient or when sampled evidence can be processed off-device.
Platform health / safety	Endpoint / asset	Stays local because safety and continuity cannot wait for network availability or upstream approval.
Trust / anomaly detection	Endpoint → node → edge	Moves upward as correlation scope increases: from one asset, to one team, to multiple teams, agencies, or jurisdictions.

FUNCTION	LIKELY HOME	MOVEMENT LOGIC
Triage / evidence ranking	Node → edge	Starts near the operator for local prioritization. Moves upward when duplicate reports, competing claims, scarce resources, or cross-team ranking become the harder problem.
Sensitive inference	Edge / governed enclave	Runs where residency, privacy, attestation, access control, and auditability can hold. The right home is determined less by model size than by governance boundary.
Coordination / deconfliction	Node → edge	Lives locally for one team. Moves to the regional edge when multiple actors share airspace, terrain, resources, routes, patients, targets, or claims on the same operating picture.
Planning / optimization	Node / edge / core	Moves upward with scope, connectivity, and time horizon. Degrades downward to cached or simplified planning when isolated.
Operator assistance	Node / edge	Remains advisory near the human decision-maker. Heavier summarization, retrieval, and scenario comparison can move upward when connected.
Policy enforcement / governance	Edge / core, with local guards	Policy is authored and evaluated centrally or regionally, but must be enforced locally enough to matter in real time.
Training / evaluation / improvement	Core	Belongs mostly in the core, where large-scale data, simulation, validation, and model evaluation live. Live operational learning should be treated as constrained and high-risk, not assumed.

Placement is not the last decision. The valley's crux was not only where the function ran, but what its output was allowed to cause: a drone could search, rank, suppress low-confidence candidates, and elevate stronger ones, but declaring the person found and sending a crew into a dark, unstable ravine remained a human command decision.

The flood exposed the same boundary at a different layer: a regional edge could merge duplicates, reconcile reports, enforce sharing rules, and recommend where scarce boats should go next, but committing another agency's crew, or declaring that a protected identity match was operationally actionable, remained with incident command.

Operational AI therefore decomposes three times: by cognitive function, by placement, and by authority. A function can observe, rank, recommend, or coordinate without being allowed to command. Deciding where it runs is not the same as deciding what authority it carries.

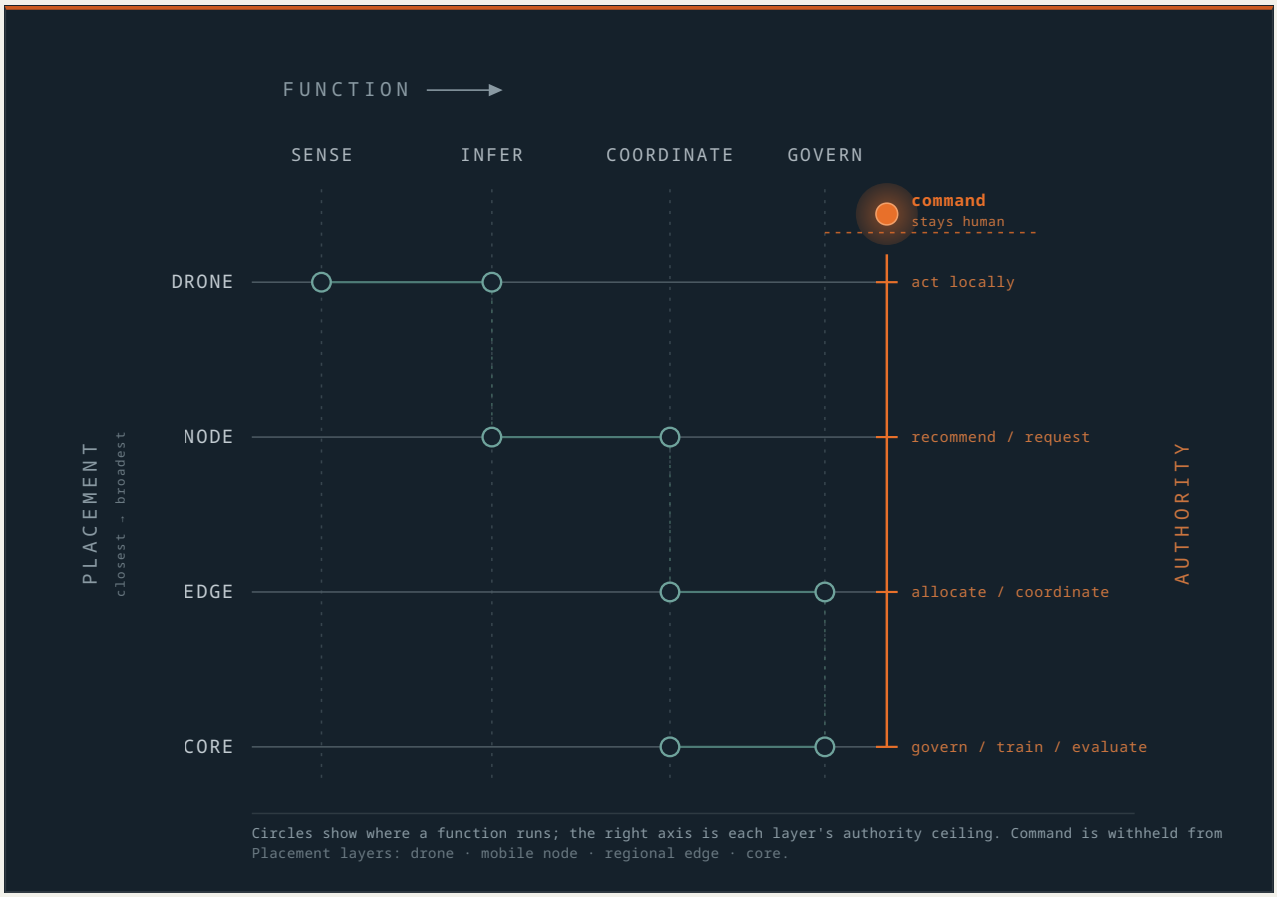


EXHIBIT 1 · THREE AXES, NOT A PIPELINE Operational AI decomposes by function, placement, and authority. Each bounded function has a natural home and an authority boundary. The mistake is treating model output, system action, and command decision as the same thing.

The sections that follow examine those decisions one at a time. Each decision turns on a constraint: bandwidth, latency, compute, safety, governance, trust, or authority. As those constraints change, the right architecture changes with them. The first decision is placement: given a portfolio of bounded functions, where should each one run?

Section 01

Placement: the four-layer cognitive substrate

The first placement mistake is treating "the AI" as one thing that needs one home. Once cognition has been decomposed into functions, that question falls apart. Perception wants to live near the sensor. Hardware health wants to live where failure begins. Triage wants fleet context. Evidence replication wants persistence. Training wants history and scale.

So placement becomes a per-function decision.

The drone is the tightest constraint: on the airframe every function shares one budget of watts, heat, accelerator cycles, memory, the per-frame window, and a draining battery, so it needs a degradation policy and a scheduler, not merely a model runtime. The question on the drone is never whether a model is good, but which cognition keeps running when the budget tightens and which yields.

That decision has four useful homes:

1. **Drone / airframe.** Highly bounded models, hard or near-real-time, the lowest power and thermal budget in the system. It runs only cognition that cannot wait and cannot leave the aircraft.
2. **Mobile rack / command node.** A rugged vehicle, trailer, or vessel that serves the local incident fleet: the aircraft, sensors, operators, and evidence flows assigned to a search area. The number of aircraft it supports is a capacity-planning input, not an assumption in this essay. Its role is to manage that fleet under local conditions, providing enough compute for heavier inference, sensor fusion, triage, policy enforcement, and evidence logging. It is the first place that can reason across multiple aircraft while still being expected to operate when disconnected from the regional edge or core.
3. **Regional operational edge.** A persistent regional layer outside the incident itself but closer than the core: a facility, network point of presence, colocated environment, or sovereign/regional compute footprint. It supports work that must outlive or coordinate beyond a single mobile node, including multi-fleet aggregation, heavier inference, policy staging, evidence replication, data residency, after-action review, and multi-agency coordination when the mission requires it. It is also where confidential inference or attested execution becomes plausible for sensitive workloads, such as missing-person matching or imagery that incidentally captures homes, vehicles, or bystanders, assuming the overhead and hardware constraints hold.

4. **Core.** Hyperscale or sovereign-cloud capacity for training, simulation, large-model reasoning, long-term analytics, synthetic data, model evaluation, and global fleet learning.

The regional edge should not be understood as simply "bigger compute." It matters because it is the first layer that can coordinate across mobile nodes while still staying close enough to the mission to act in time. It is also the first plausible place to run larger bounded models reliably under mission constraints: with enough compute to serve heavier inference, enough shared context to make caching useful, and enough governance machinery to support confidential or restricted inference at scale.

At that layer, two different substrates start to separate. One is the inference-serving substrate: running bounded models, routing requests, scaling capacity, observing performance, caching repeated context or token prefixes where the serving stack allows it, and rolling back bad behavior. The other is the agent-operating substrate: authenticating agents and tools, carrying bounded mission context, enforcing policy, exchanging claims, preserving provenance, and emitting audit events.

Those are related, but they are not the same problem. The inference-serving substrate answers where and how a model runs. The agent-operating substrate answers what a model, agent, or tool is allowed to touch, remember, call, recommend, or cause. Collapsing them back into "the AI model" repeats the mistake this essay is trying to avoid.

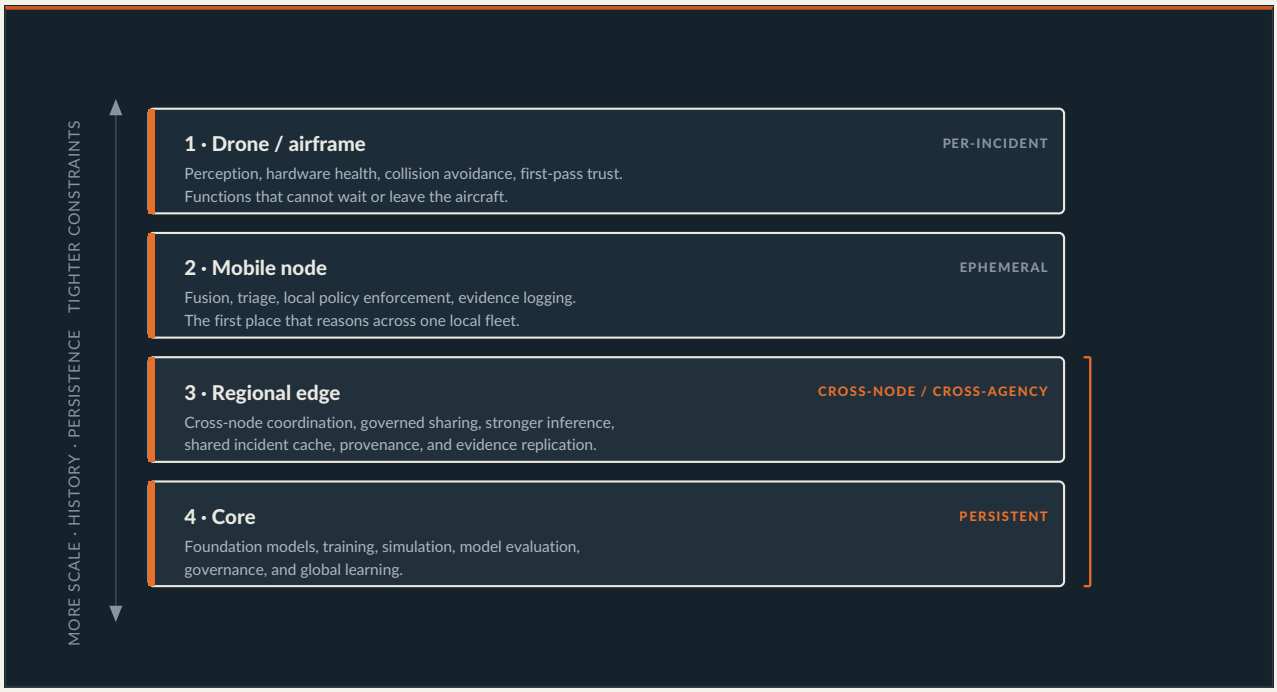


EXHIBIT 2 · THE FOUR-LAYER SUBSTRATE Each layer hosts functions the others cannot. Budget and latency tighten toward the airframe; scale, history, and persistence grow toward the core. The mobile node is local to one incident fleet. The regional edge is where cross-node coordination, governed sharing, and shared incident context become possible.

The tension. Different functions pull toward different layers. Perception, health, and first-pass trust pull down to the airframe because latency and available connectivity remove the alternatives. Fusion, triage, coordination, and policy pull to the mobile node because they need fleet context and must survive disconnection. Aggregation, residency, and cross-agency work pull to the regional edge. Training and whole-area learning pull to the core.

What breaks the tie. Physics decides the lower layers; scope decides the upper ones. Available connectivity, latency, heat, and power decide what stays on the aircraft. Disconnection decides what the node must do alone. Agency span, residency, and cross-incident history decide what belongs at the regional edge. Scale and validation decide what stays in the core.

Residual cost, and the unresolved constraint. Keeping cognition on board means accepting smaller models and harder fleet operations. The decision flips only if connectivity changes category: a persistent high-capacity, low-latency connection over the whole search area would move perception to node or edge models and leave the aircraft with a pre-filter. Actual connectivity across the search area decides this fork.

Section 02

On-board model: size vs accuracy

Once perception lands on the airframe, the next question is how much model the aircraft can actually carry, a sustained-operation question, not a benchmark one: what can run hot, on battery, while the sortie is still underway? A bigger on-board model perceives better; a smaller one survives the aircraft.

Pulls. Toward bigger: detection quality is the difference between cueing a human and missing a person, and the error is asymmetric. Toward smaller: compute, power, and the thermal envelope cap sustained inference, and accuracy degrades as the device heats, so the benchmark number is not the field number.

What breaks the tie. Sustained thermal performance. The question is not whether the model runs when the hardware is cool, but whether it still holds the required frame rate and detection quality after the compute module, camera, radios, and flight systems reach operating temperature. If a larger detector throttles mid-sortie, it stops being the better model in practice. The airframe should run the largest detector that stays inside its power and thermal budget, and push task-relevant features upstream so heavier second-look inference can happen at the node.

Residual cost, and the unresolved constraint. We accept lower single-frame accuracy and lean on overlapping passes, node-side second looks, and human confirmation to recover some misses. The unresolved constraint is sustained frame-rate-at-accuracy on the chosen airframe and accelerator, hot. If the airframe can sustain a useful detector under those conditions, it runs real perception locally. If it cannot, the airframe degrades to a motion-and-heat cue generator, and full detection moves node-side.

Section 03

Multipurpose small model vs bounded specialized models

Once the airframe has a finite inference budget, the next fork is composition: one small multipurpose model, or several bounded models with narrower jobs.

> *The system’s real priorities show up under constraint.*

Option A: one multipurpose small model. Simpler to deploy, with shared context and some ability to generalize, but harder to certify, bound, schedule, and degrade. It may also spend scarce compute on work that a specialized model, classical estimator, or rule would do faster and more predictably.

Option B: distinct bounded models. Easier to test, isolate, schedule, and degrade gracefully, and friendlier to a safety case, at the cost of integration complexity, version sprawl, and more Day-2 burden.

Option C: hybrid, and the recommendation. Use bounded models, and classical methods where they are better, for perception, hardware health, positioning integrity, and trust/anomaly, the functions that need predictable latency, isolation, and a defensible safety case. A language-capable or multimodal generalist is optional, not assumed. If the airframe has enough compute, power, and thermal budget for one, reserve it for advisory work only: summarization, triage support, and explanation of system state. It is not flight-critical, cannot retask the aircraft, and cannot override the system's authority, privacy, evidence, or safety rules.

The hybrid is not a fence-sit. The system's real priorities show up under constraint. That is where the classic difference between needs and wants becomes visible. When the local budget is comfortable, many arrangements appear to work and the choice can look aesthetic. When watts, heat, memory, or frame budget tighten, the question stops being "which AI is best" and becomes "which cognition is allowed to keep consuming the scarce local budget." Those become scheduling and degradation decisions. Bounded models make that expressible; one fused model makes it opaque.

A compact degradation policy makes the point concrete. The table is not a tuned configuration; it is a way to show which functions yield first under pressure:

CONDITION	KEEP LOCAL	DEGRADE OR OFFLOAD
Normal	perception, health, trust, positioning	nothing local sheds; node handles triage and planning
Thermal pressure	health, trust, positioning, reduced-rate perception	drop perception frame rate; defer non-urgent inference
Low battery	health, positioning, minimal perception	shed the advisory model first, then trim perception to keep flight-safe cognition
Link degraded	the full local portfolio	hold detections locally; send only task-relevant evidence when the link returns
Candidate found	perception, positioning, evidence capture	prioritize the find: raise perception fidelity on the track, queue everything else
Sensor suspect	trust/anomaly, positioning cross-check	quarantine the suspect stream; do not propagate its observations upward

That table provides the shape of an operating doctrine. Under pressure it shows which functions are load-bearing: hardware health, positioning integrity, and trust checks. It also draws a clear line: reflex is the last to yield, and advisory assistance yields first.

EXHIBIT 3 · WHAT YIELDS UNDER SCARCITY The airframe's shared budget forces a priority order. Advisory assistance and non-urgent inference shed first; perception degrades before safety, positioning, health, and deterministic failsafes. The degradation order is part of the system's operating doctrine.

Section 04

Coordination: the audit backbone, and the partition fork

Coordination and partition are one fork seen from two sides: reconciling two crews that report the same rooftop, and keeping a cut-off node working alone.

We can skip the generic request/response versus event-fabric debate; for this system, the deciding constraint is audit. A public-safety system taking autonomous or semi-autonomous actions has to reconstruct what each agent did, what it observed, what it believed, and why it acted. An event log provides that replayable trail by design. Events become the backbone; requests become the in-path exceptions.

The second tie-breaker is what happens when coordination breaks. In distributed-systems terms, this is a partition: two parts of the system are still running, but they cannot reliably talk to each other. In this paper, the vulnerable layer is the mobile node, the incident-local system that sits between the aircraft and the regional edge. It is designed to keep working when disconnected, which means it is also designed to become an island.

A mobile node may lose connection to the regional edge, or two mobile nodes working the same incident may lose connection to each other. At that point, search-cell ownership has to resolve without access to the normal coordinating authority, whether that authority is shared incident state, a regional coordination service, or a human command function reachable through the system. Who is responsible for sweeping this cell? What happens to a candidate found in it? What if the authority that would normally deconflict cell ownership, candidate status, or retasking is unreachable?

The safe default is static pre-partition assignment: each mobile node owns its cells, holds its finds, and reconciles on reconnect. That is a consistency-versus-availability choice, and search and rescue should bias toward availability. Keep sweeping the cell you own on stale information rather than stall for a coordinator that may never answer. The multi-UAV search literature keeps returning to coordination under partial connectivity for a reason: autonomous systems do not remove the old SAR coordination problem; they make it explicit in software. Standard SAR practice already depends on assigned areas, accountable tasking, and later reconciliation. The design choice is how much of that operational discipline the AI system preserves when the network breaks.

Section 05

The trust model: Zero Trust as evidence control

Authenticate the drone all you like; it does not make the observation true. That is the failure mode the trust layer exists to address, and the reason Zero Trust here has to mean evidence control.

> *A signed false observation is still false.*

Trust has to be scoped and revocable. In this system, a drone is not trusted in general. It is trusted for a specific mission, geofence, time window, software version, sensor package, connectivity mode, and autonomy level, and only when a particular observation clears the required corroboration threshold. Remove one of those conditions, and the same authenticated drone may no longer be trusted for the action in front of it.

That resolves into four questions, not one:

TRUST QUESTION	WHAT IT GOVERNS	WHY IT MATTERS
Entity trust	Is this drone, node, or operator who it claims to be?	Prevents spoofed actors and unauthorized tasking
Behavior trust	Is it acting within mission and policy bounds?	A valid drone can still behave wrongly: hijacked, misconfigured, or steered by a bad input
Data trust	Can we trust the sensor, coordinate, timestamp, and model output?	A signed false observation is still false
Decision trust	Is this action allowed without more human review?	Determines escalation, retasking, and asset commitment

Entity trust is the easy one, and where many designs stop. The other three are the mission. Behavior trust asks whether the drone is acting within its assigned bounds. Data trust asks whether this particular sensor reading, coordinate, timestamp, and model output are reliable enough to use. Decision trust asks whether the resulting action is allowed without more human review. The point is not whether the drone is authenticated in the abstract, but whether this observation, from this sensor, on this software version, at this coordinate, is good enough to trigger the next action.

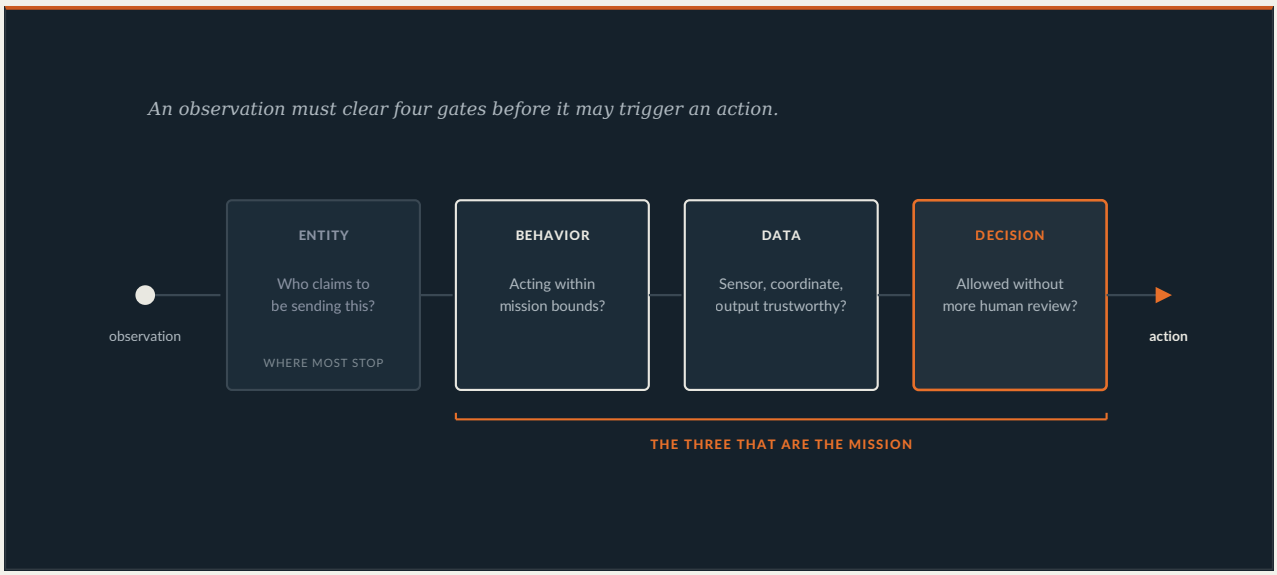


EXHIBIT 4 · TRUST IS EVIDENCE CONTROL Authentication answers only the first gate. The other three, behavior, data, and decision, are the mission: whether a particular observation is good enough to trigger the next action.

The same four questions apply beyond drones and their observations. On the agent-operating substrate, an agent calling a tool or an API is subject to exactly this scoped, revocable trust: who is calling, what authority it holds, what data it is allowed to see, what action it is allowed to trigger, and how the decision is logged. A tool call is an observation with consequences, and authenticating the agent no more makes its request safe than authenticating a drone makes its observation true. A claim exchanged between agents, a governed API call, and a sensor reading all clear the same gates before they are allowed to act, and the audit record is part of the action, not an afterthought.

Two mechanisms make that trust model enforceable. The first is message integrity. Drones sign telemetry and tasking so the mobile node can validate claims without a round trip to the core. Request-signing on this medium is a design **assumption** here, not established practice, but the underlying pattern is familiar: validate the message, bind it to an identity and context, and reject claims that fall outside the mission scope. Deterministic circuit breakers such as geofence, return-to-home, and disarm still bound misbehavior regardless of what the message says.

This approach is not exotic. Cross-domain federated identity has long existed in public-safety information sharing, and aviation already requires Remote ID broadcast. But those mechanisms only get partway there. Remote ID says who and where; it does not say whether an observation is trustworthy or whether an action is authorized. That is the gap the trust layer has to close.

The second mechanism is execution integrity. When the trust scope names a software or model version, remote attestation checks that claim rather than taking it on faith. An attested environment can prove that the drone or node is running the expected, untampered software before its observations are allowed to count.

Policy is defined centrally but enforced in the path, because the core may be unreachable when the decision has to be made. The drone enforces immediate safety bounds. The mobile node validates messages, correlates behavior across the fleet, and applies local policy while disconnected. The regional edge federates policy across agencies or incidents when connected. The core remains the authority for identity, policy, model governance, and audit, but it cannot be the only place enforcement lives.

Section 06

Data Governance: what may be sensed, stored, shared and learned

Search and rescue sensors do not only capture the thing being searched for. A drone looking for a missing hiker may also see rescuers, law enforcement, homes, vehicles, license plates, or people who have nothing to do with the incident. Some of that context may matter; much of it should not propagate just because it was captured.

That is why governance has to begin at capture, not later in the core. The first decision is not simply whether the system detected something, but what the observation is allowed to become: evidence, context, shared data, retained data, or training data. A frame that plausibly relates to the search may need to be retained. A face, plate, or private property detail that does not meet that bar may need to be blurred, minimized, dropped, or kept under a tighter evidence rule. If the core makes that decision after the raw frame has already left the aircraft, governance has arrived too late.

Governance is the other side of the trust plane. Trust decides which claims may propagate; governance decides which data may propagate, persist, and later train the system. It is a real decision point, not just a logging policy, because the answer changes with conditions. Strict privacy or jurisdiction rules push inference and minimization to the drone, or as close to it as compliance and capability allow, with only task-relevant evidence moving up. Confidential-inference hardware and attested execution can make heavier processing off the core more defensible, though whether the overhead holds on a rugged mobile node is still a measurement, not an assumption. A unified incident authority with clear retention permissions can replicate more data regionally for review and training. Degraded connectivity also forces governance to be enforceable offline, in the drone and mobile node.

This is where the agent-operating substrate earns its place. Access, retention, minimization, and sharing rules are not policy documents to be checked later in the core; they have to be executable in the live path, enforced at the moment an agent reads, moves, or shares something. The default exchange between agents is a bounded claim and its task-relevant evidence, carrying provenance and access markings, not raw data. Raw collection moves only when a rule explicitly allows it. That is what keeps minimization and residency real when the network is up and several parties are sharing at once, rather than aspirations that hold only until the first fast, convenient copy.

Three control points make those decisions operational. The first is sensor provenance: a detection from a sensor the health model just flagged is weaker than the same detection from a clean one. The second is the learning boundary: the path from a captured frame to a training set is a governed decision, the same control point as Section 9. The third is

offline enforceability: buffered evidence has to reconcile with regional and core records on reconnect without losing chain of custody. Retention windows and jurisdictional lines are not after-the-fact paperwork; they decide what may persist, where it may live, who may see it, and whether it may become training data.

Section 07

Positioning when the absolute reference is unavailable or wrong

A search system has to place what it sees well enough for someone else to act on it. In search and rescue, that means positioning is part of the evidence chain, not just navigation metadata. A candidate detection, a sensor reading, and a coordinate travel together; if the coordinate is wrong, the rest of the evidence may still be true but operationally misleading.

The tension. A "find" is only actionable if its location can be trusted. A satellite fix supplies that cheaply, until it is jammed, shadowed by terrain, or spoofed into a confident wrong value.

Pulls. Toward trusting the fix: it is accurate, ubiquitous, and cheap. Toward a cross-check: a wrong coordinate is more dangerous than a missing one, because it sends rescuers to the wrong place certain they are right.

What breaks the tie. Integrity, not availability. The system cannot treat one source as ground truth when it can be silently corrupted, so it needs an independent cross-check: visual-inertial or terrain-relative methods, time-of-arrival multilateration, surveyed anchors. The coordinate carries its provenance, per Section 6.

Residual cost, and the unresolved constraint. The achievable accuracy of a non-satellite positioning stack under degradation decides the rest of the chain, and the essay does not have it. Hold to tens of metres and the system produces an actionable point a human can commit assets to. Drift to hundreds and a "find" becomes "re-search this area," a sweep rather than a pinpoint recovery, and the asset-commitment decision in Section 8 changes with it. The fork flips back to trusting the fix only in benign conditions.

Section 08

The autonomy ladder, and the boundary that stays human

In a real incident the AI does not command the mission. It attaches to the incident-command structure as a subordinate, bounded from outside by delegated authority, airspace control, the connectivity it has, retention rules, and the workflows of the ground, aviation, medical, and law-enforcement teams already on scene. So the design question is not only where cognition runs but where its output is allowed to become a command, and that line is drawn by the command structure, not the model. This is the third decomposition the portfolio implied: by function, then placement, then authority.

> *A human who is accountable and equipped to judge is doing oversight; a human clicking approve down a queue is decoration.*

The error-cost asymmetry sets the levels. A missed person can be fatal; a false positive is a wasted sortie. So the system is tuned to surface more candidates and act faster within bounds, up a short ladder: Shadow (model watches, humans act), Supervised (model proposes, humans approve), Bounded (the aircraft searches and tracks autonomously inside a geofenced, time-boxed envelope). What stays human is everything irreversible or cross-boundary: declaring a located person, committing a crewed asset, re-tasking across agencies.

That boundary is narrow on purpose, and not only because operators are scarce. A human asked to approve everything becomes a rubber stamp, what one critique calls liability laundering, and the evidence is blunt: regulators name automation bias as a hazard the overseer must resist, and studies show reviewers tend to inherit a model's errors rather than catch them. So the human is reserved for the decisions where accountability genuinely attaches, and those are engineered to be judgeable, with the evidence, the provenance, and the reason a candidate surfaced placed in front of the operator rather than a bare yes-or-no on a number. A human who is accountable and equipped to judge is doing oversight; a human clicking approve down a queue is decoration.

Flip the asymmetry, into a setting where the irreversible act is harmful rather than helpful, and the same ladder tunes toward restraint. The human sits in the same seat; only the threshold and the direction of acceptable error change.

Section 09

Central retraining vs in-field learning

The model drifts with season, terrain, light, and pose. Adapting in the field is faster; adapting centrally is safer.

Pulls. Toward in-field: drift is continuous, labelled data is scarce, and methods exist that keep raw data on the device. Toward central: a perception model that decides where to send rescuers is safety-relevant, and you cannot validate a model in the field.

What breaks the tie. The inability to validate in the field against an asymmetric error cost. An unvalidated update that quietly lowers recall is, in the moment, indistinguishable from ordinary misses, and the cost is a missed person. So learning happens centrally with staged rollout, through the same governed capture-to-training path as Section 6. In-field continuous learning for this role is **speculative**.

Central does not have to mean centralized data, though, and that is the part the plain framing hides. Federated adaptation is a distinct middle path: the fleet computes model updates locally and ships gradients rather than raw frames, a central process aggregates and validates them, and only a validated model is pushed back. It keeps the validation gate the error-cost asymmetry demands while honoring the same data-governance boundary as Section 6, and it is established practice rather than speculation. So the real line is narrower than "central versus in-field." What stays speculative is letting a field-trained update touch live perception without central validation.

Residual cost, and when it flips. You run a staler model than the lab has, with a retrain-validate-push lag set by the update cadence. It flips toward bounded in-field adaptation only when both hold: an in-field evaluation that can certify a candidate update before it touches live perception, and enough connectivity to supervise it. The regime grants neither.

Section 10

Day-2: operating a graph of models, policies, and device states

Placement is a moment; operating the fleet is forever. Once cognition is a portfolio, Day-2 is not "push a detector to drones." It is operating a graph of models, policies, evidence schemas, runtimes, device states, trust scopes, agent permissions, bounded context, and audit rules.

By graph, I mean a set of operated objects connected by contracts. A perception model emits an output schema that triage consumes. A policy version determines what an agent may call. A trust state determines whether a claim can propagate. An audit rule determines what must be logged. Each object changes on its own clock, has its own owner, and rolls back in its own unit.

Naming the objects is what turns "operate the fleet" into something someone can own:

OPERATED OBJECT	UPDATE CADENCE	ROLLBACK UNIT	LIVES / ENFORCED AT	OBSERVABILITY SIGNAL
Perception model	Per sortie, staged across the fleet	Last-good model, this function only	Drone	Frame-rate-at-accuracy hot, drift
Hardware-health model	Rare, conservative	Last-good, per function	Drone	Missed-fault and false-alarm rate
Trust / anomaly model	As threats and spoofing evolve	Revert without grounding the fleet	Drone, then node	Out-of-bounds rate, spoof catches
Coordination model	As fleet size and tactics change	Node-local revert	Node	Cell-ownership conflicts, dedup accuracy
Policy (trust + governance)	On any authority, jurisdiction, residency, or sharing-rule change	Versioned, independent of model pushes	Decided core, enforced in-path at drone/node/edge	Policy lapses, denied actions, residency breaches, offline-enforcement gaps
Evidence schema	Rarely, with care	Must stay backward-readable to consumers	Spans all tiers	Schema-mismatch errors between producer and consumer
Device / firmware state	Per maintenance cycle	Per device	Drone	Battery, thermal, sensor, positioning health

OPERATED OBJECT	UPDATE CADENCE	ROLLBACK UNIT	LIVES / ENFORCED AT	OBSERVABILITY SIGNAL
Trust state (who is in or out of bounds)	Continuous, at runtime	Not an artifact; it is live state	Node, with local enforcement	Currently revoked or quarantined entities
Inference runtime (serving substrate)	On serving-stack changes, apart from models	Runtime version, without touching model weights	Node, edge, core	Throughput, tail latency, queue depth, routing hit-rate
Cache and routing policy	On serving-topology or reuse-pattern change, apart from models	Routing rules and cache tiers, without touching weights	Node and edge, where the shared incident cache lives	Cache hit-rate, tail latency, stale-context serves
Attestation and key-release	On software or measurement change, and on key rotation	Prior measurement set and release policy	Edge and node, where sensitive inference runs	Attestation failures, key-release denials, unmeasured binaries
Agent / tool gateway (agent-operating substrate)	On tool, API, or workflow change	Per gateway or per tool binding	Node and edge	Denied and re-allowed calls, tool error and timeout rates
Agent identity and authorization policy	On authority, delegation, or scope change	Versioned, apart from model and runtime pushes	Decided core, enforced in-path	Failed auth, scope escalations, revoked agents
Bounded mission context / memory	Per mission and session	Bounded and expirable, not a durable artifact	Node and edge	Stale-context use, retention-window breaches
Claim schema	Rarely, with care	Must stay backward-readable to consumers	Spans node, edge, core	Claim-parse failures between agencies
Audit / event-log policy	On governance or retention change	Versioned	All tiers	Missing or unverifiable events, reconciliation gaps

The rows update on different clocks, roll back in different units, and live at different layers. That is why Day-2 is a graph problem.

Two operated objects are especially easy to collapse into "the runtime," and they should not be: the inference-serving and agent-operating substrates from Section 1 change independently. A model rollback does not touch an authorization policy. A tool-gateway

change does not touch model weights. Each emits different signals: throughput, latency, queue depth, and routing hit rate on one side; denied calls, scope escalations, failed claims, and unverifiable events on the other. Watching only the first repeats the "AI model" mistake.

One placement error is worth making explicit. The familiar inference optimizations, continuous batching, paged attention, prefix caching, and prefill/decode splitting, are most useful where many requests share accelerators and context. That is the node, regional edge, or core. An airframe running one stream under tight power and thermal limits gets little from those techniques. Its real levers are smaller models, quantization, scheduling, degradation policy, and thermal management.

The operating tie-breakers are short. Updates roll out by sortie, to a fraction of the fleet, per function, with a last-good rollback. Planning and coordination stay warm because a cold start can consume the golden hour; non-urgent training and analysis can scale down. Observability is eventually consistent because it has to be: what must act in the moment runs locally, telemetry and event logs buffer locally, and reconciliation on reconnect preserves chain of custody.

> *The standing cost is operating the graph, not the drones.*

Two Searches, Reconstructed

We opened with two searches. Now we run them back through the framework. The portfolio of functions is the same, the four placement layers are the same, the trust questions are the same, and the human boundary on irreversible command is the same. What changes is the constraint set. Change connectivity and scope, and the same framework produces a different operational shape.

Each search below follows the same pass: the constraints it faces, where placement lands, how trust and governance resolve, where authority stays, and what changes the design. The result stays illustrative rather than specified, and qualitative where the essay lacks a firm constraint. The point is the contrast.

Missing Hiker

Constraint set. The tower is gone, so the regional edge and the core are both unreachable. The survival clock is short, and the drone fleet is bounded by battery, compute, and thermal headroom. The operating question is local triage under scarcity, with no one upstream to ask.

Placement. The system collapses inward. Perception, hardware health, first-pass trust, and positioning stay on the drone because they cannot wait and cannot leave the aircraft. Fusion, triage, false-positive suppression, evidence logging, and operator assistance move to the mobile node, the only layer that can reason across the local fleet. The regional edge and core fall out of the live loop.

This is the partition from Section 4 made real: the node is an island by design. It keeps sweeping the cells it owns on stale information rather than stall for a coordinator that will never answer.

Trust and governance. The drone publishes candidates, not truths. Each candidate carries its confidence, sensor state, model version, coordinate provenance, and the local policy state that decides what may be retained or escalated.

The mobile node becomes the local trust layer. It correlates passes across its own aircraft, suppresses obvious false positives, heat off sun-warmed rock, glare off water, a deer bedded down, checks sensor health and positioning integrity, and decides which few candidates deserve human review. Only task-relevant evidence is held or escalated. Nothing raw propagates just because it was captured.

Authority. Reversible actions run locally on the fleet's own authority: re-image the candidate, tighten the track, sweep the next cell. Being wrong costs battery and time. Routing every one of those actions to an approval click would manufacture the decision fatigue that lets the real find slip past.

Irreversible action stays human: declaring a located person and committing a crew down into a dark, unstable ravine. That is where accountability attaches and where the operator has to be equipped to judge. Placement collapsed inward. Command authority did not.

Placement collapsed inward. Authority did not move at all.

What changes the design. If the airframe cannot sustain frame-rate-at-accuracy while hot, full detection moves to the node and the drone degrades to a cue generator. If positioning integrity degrades from a usable point to a broad uncertainty area, a "find" becomes "re-search this zone," and the asset-commitment decision tightens. If reliable low-latency connectivity returns, second-look inference and broader correlation can move upward. The architecture is local because this constraint set leaves no better home.

Multi-Agency Search

Constraint set. County, state, and federal teams operate inside different authority domains, each with its own aircraft, operators, incident-local nodes, evidence rules, and access controls. Regional connectivity holds well enough that a federation layer stays reachable. Search areas overlap, the same candidate may be seen by more than one fleet, and some imagery captures homes, vehicles, bystanders, or missing-person records.

The operating question is cross-agency interpretation, fast, under rules that do not relax because the river is rising.

Placement. The system expands outward. First-pass perception still starts on the aircraft, and each node still controls its own fleet, but the live work moves to the regional edge. It is the only layer that sits above any single domain, close enough to close the loop in time, and able to keep residency and attestation near the mission.

A mobile node cannot be the referee because it is bounded to one fleet and one authority domain. The core cannot hold the loop either because the decision path is too slow and too distant for a rising river. The core stays out of the live loop; training, simulation, and long-term evaluation happen later.

Trust and governance. Agency-facing agents exchange claims through governed interfaces, not raw data. The regional edge correlates claims across fleets, deduplicates sightings, reconciles overlapping cells, applies visibility rules so each agency sees only what its authority permits, and records why every claim was shared, withheld, downgraded, or escalated.

Sensitive work, such as matching a candidate against a missing-person record or screening imagery that caught bystanders, runs inside an attested environment where the overhead allows. This is the trust model from Section 5 operating between organizations rather than between a drone and its node. Identity is necessary, but the real work is evidence control across domains.

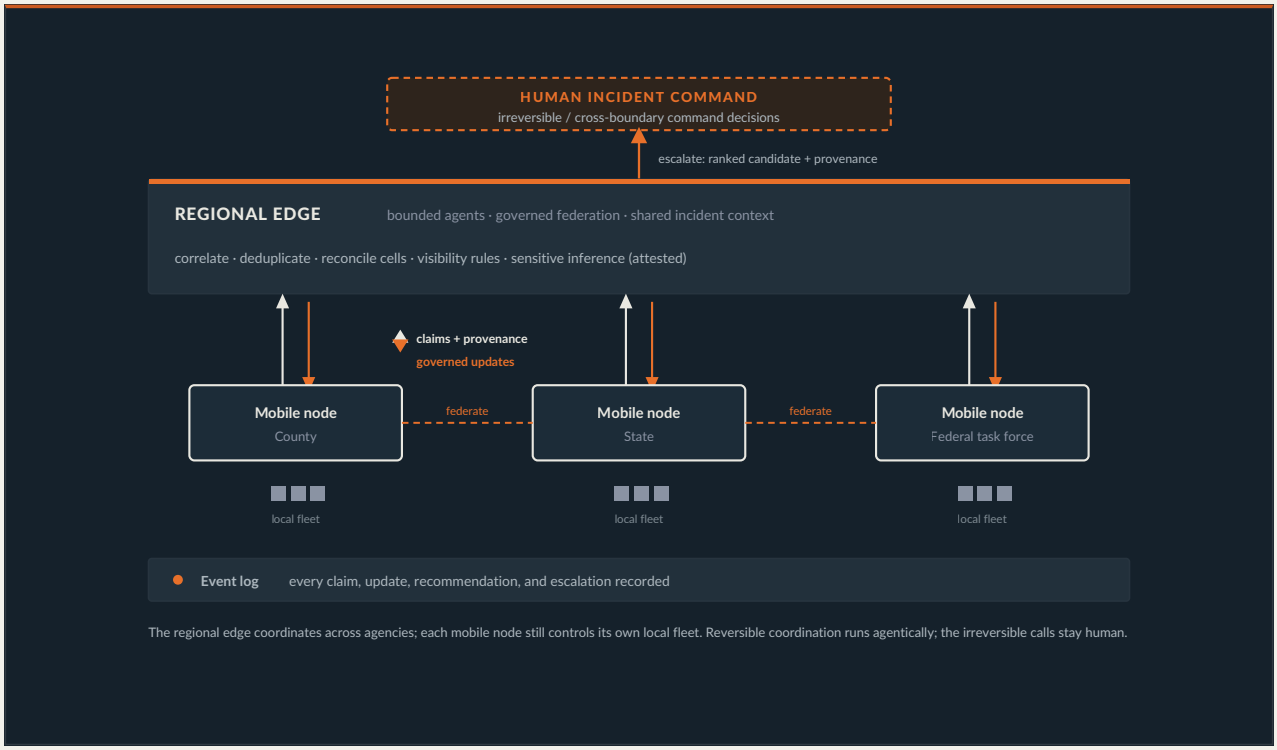


EXHIBIT 5 · GOVERNED FEDERATION AT THE REGIONAL EDGE In the flood case, claims and provenance flow up from each agency's mobile node. Governed updates and recommendations flow back down, filtered by each agency's visibility rules. Federation also runs sideways across authority domains. Bounded agents reconcile, coordinate reversible updates, and escalate irreversible cross-boundary decisions to human incident command. Every exchange is logged.

Authority. What returns to the nodes is governed updates, not commands: one node learns its candidate is likely a duplicate, another that a cell needs a second look, a third that a track is shareable for response but not for broad retention. Each node translates those updates into local tasking for its own fleet, inside its delegated authority.

The cross-boundary calls stay with incident command: committing a crewed asset, retasking another agency, or declaring a located person. The regional edge coordinates

and informs. It does not own the final command, which is exactly why it can move quickly without requiring a human to reconcile every event by hand.

What changes the design. If regional connectivity drops, coordination degrades toward the node, and each fleet falls back to working its own cells, as in the hiker case. If a single accountable command structure with clear retention authority forms, more evidence can replicate regionally for review and training. If confidential-inference overhead proves too heavy on a rugged regional deployment, sensitive matching narrows or waits for the core. The architecture is regional because this constraint set puts the load-bearing work above any one agency and below the distant core.

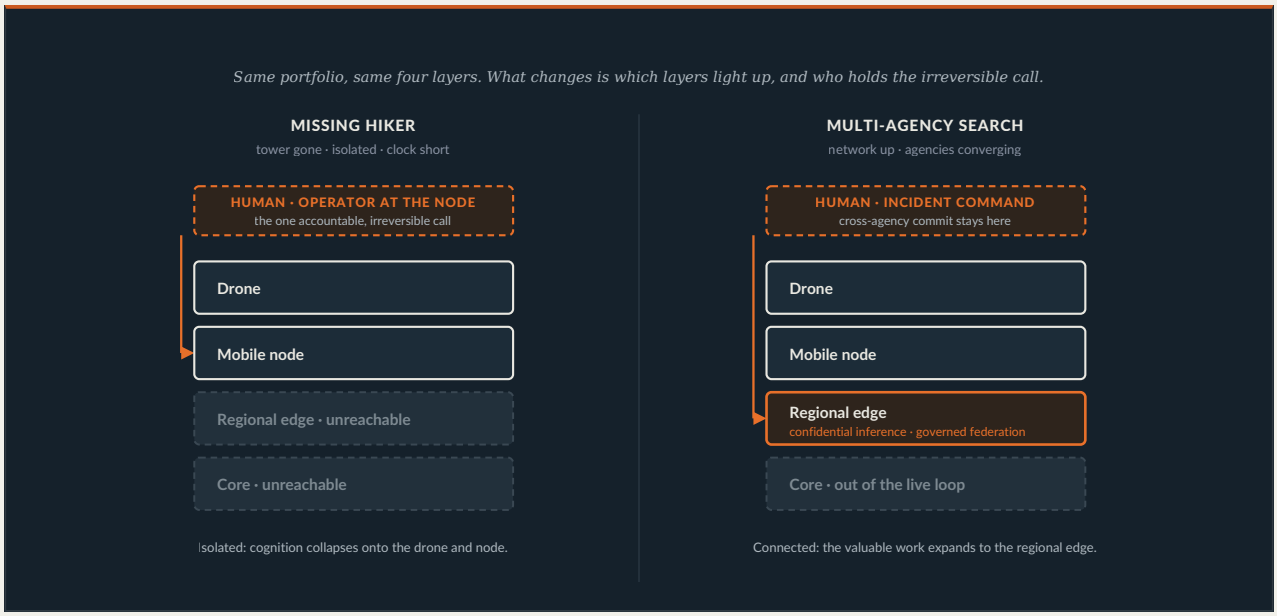


EXHIBIT 6 · THE SAME MACHINERY, TWO REGIMES Missing Hiker and Multi-Agency Search run the same portfolio and the same four layers. Isolated, the work collapses onto the drone and node, and the operator holds the one irreversible call. Connected, it expands to the regional edge for confidential, governed coordination, while incident command holds the cross-agency call. In both, the core stays out of the live loop. The parts barely move; the seams carry the difference.

The Constraint Map

The constraint map is the artifact of the reasoned exercise. It is not a proposed architecture, a drone-fleet design, or a bill of materials. The two searches serve as forcing functions, but the map is broader than either scenario: it identifies where changes in connectivity, latency, compute, trust, governance, human attention, or error cost change the right answer.

It should not be a giant architecture diagram. It should be a compact decision table: when a constraint tightens or relaxes, a design decision flips. Each row also names the axis it most moves, and the pattern is telling. The set of functions is largely fixed by the problem, so the constraints rarely move function; they move where functions run (placement) and what their outputs are allowed to cause (authority).

CONSTRAINT	IF IT TIGHTENS	IF IT RELAXES	AXIS	DECISION AFFECTED
Connectivity	Move inference, triage, and planning down toward drone and node	Move coordination, deduplication, and shared context up to edge and core	Placement	Where inference, triage, and planning run
Latency	Keep perception, safety, and urgent triage local	Allow heavier inference, retrieval, and scenario comparison upstream	Placement	Local vs upstream; model size
Compute and thermal budget	Shed advisory functions first, reduce frame rate, prefer bounded models	Run heavier models, richer fusion, broader search over context	Function	Function composition; degradation order
Trust and provenance risk	Require signed observations, audit trails, corroboration, and anomaly checks	Permit faster sharing and lower-friction coordination	Authority	Trust / evidence control
Jurisdiction and data rules	Keep sensitive inference regional or local, restrict sharing	Allow broader aggregation and centralized analysis	Placement	Data residency and sharing scope
Human attention	Suppress low-confidence noise, rank evidence, escalate fewer items	Allow more review, richer explanations, slower deliberation	Authority	Escalation threshold and autonomy
Error-cost asymmetry	Limit autonomy, require human command for irreversible actions	Permit more autonomous local action when the failure cost is bounded	Authority	Autonomy vs the command boundary

CONSTRAINT	IF IT TIGHTENS	IF IT RELAXES	AXIS	DECISION AFFECTED
Validation availability	Freeze live learning, rely on prevalidated models and logs	Allow controlled updates, evaluation, and improvement loops	Authority	Learning / model lifecycle
Positioning integrity	Cross-check location, degrade confidence, avoid precise tasking	Allow more direct route planning and evidence localization	Authority	Positioning trust and actionability
Mission scope	Optimize for one crew or fleet	Move toward regional federation and shared incident context	Placement	Federation and coordination scope

Findings

The main finding is that the model is not the stable unit of analysis. The bounded function is. A detector matters, but it is only one function in a larger portfolio, and it is useful only inside a contract: a latency budget, a thermal envelope, a frame-rate requirement, an output schema, a confidence calibration, a provenance trail, and a defined failure mode. A better benchmark does not automatically produce a better operational system; if a new detector breaks the contract, the surrounding system has to change with it.

The model is not the stable unit of analysis. The bounded function is.

Placement, in turn, is a per-function decision, not an architecture slogan. Perception wants to live near the sensor. Health and failsafe logic want to live where failure begins. Triage wants fleet context. Cross-agency coordination wants the regional edge. Training, simulation, evaluation, and long-cycle learning want the core. The right answer changes when the constraints change.

Authority is separate from placement. A function can observe, filter, rank, recommend, coordinate, allocate, or enforce policy without being allowed to command. The drone can make bounded local decisions, such as safety maneuvers or evidence filtering. The regional edge can coordinate, deconflict, cache shared incident context, and enforce sharing rules. But declaring a person found, committing another agency's crew, or making any irreversible cross-boundary decision remains with human command.

The regional edge, in particular, is more than bigger compute: it is the first layer where cross-node coordination, governed sharing, shared incident context, confidential or restricted inference, and meaningful caching become possible at mission scale. It is also where the inference-serving substrate and the agent-operating substrate begin to separate: one answers how bounded models are served, routed, observed, cached, and rolled back; the other answers what agents and tools may touch, remember, call, recommend, or cause.

Trust is evidence control, not just access control. Authenticating a drone, an agent, or a tool does not make its observation true or its action safe. The system has to evaluate the entity, the behavior, the data, and the decision before a claim may propagate or trigger action. A signed false observation is still false.

Finally, Day-2 is the real operating burden. Once cognition is decomposed into functions, the system is no longer operated as one model rollout; it becomes a graph of models, policies, schemas, device states, trust states, runtime versions, tool gateways, bounded context, claim formats, and audit rules, each on its own clock and rolling back in its own unit.

The constraint map turns into an implementation agenda when the missing values become measurements. A team would not start by arguing whether the architecture should be drone-first or core-first; it would start by measuring the constraints that decide those forks. Three move the largest parts of the design: sustained detector performance hot, non-satellite positioning drift under degradation, and operator bandwidth under load; Appendix A lays them out as the first test plan.

The useful result is the constraint map. The system is not defined by the model it uses. It is defined by the functions it decomposes, the constraints it survives, and the authority it refuses to pretend the model owns.

Appendix A: Constraint Measurement Backlog

The essay's constraint map identifies where design decisions move when conditions change. This appendix names the measurements an implementation team would need before turning that map into a physical architecture.

The purpose is to identify the values that decide the forks: what stays on the drone, what moves to the mobile node, what belongs at the regional edge, what can wait for the core, and where human command remains required.

MEASUREMENT	WHAT MUST BE MEASURED	DESIGN FORK IT DECIDES	EFFECT ON THE DESIGN
Sustained detector performance, hot	Required frame-rate-at-accuracy after the compute module, camera, radios, and flight systems reach operating temperature.	Whether perception runs on the drone or moves to the mobile node.	If sustained performance holds, the drone runs real perception. If it does not, the drone degrades to a cue generator and full detection moves node-side.
Airframe power and thermal envelope	Available watts, thermal headroom, battery impact, and throttling behavior under continuous inference.	Model size, degradation order, and which functions can remain local.	Tight headroom favors smaller bounded models, quantization, reduced frame rate, and shedding advisory functions first.
Candidate volume under realistic terrain	Number of false, weak, duplicate, and unresolved candidates produced per sortie under dusk, canopy, smoke, and heat.	Human review load, triage thresholds, and suppression logic.	High candidate volume pushes more filtering to the drone and node before human review. Low volume allows more direct operator adjudication.
Operator cognitive bandwidth under load	How many candidates an operator can judge under fatigue, split attention, time pressure, and radio and task load before oversight degrades.	Autonomy boundary for reversible actions and escalation threshold for human review.	Lower bandwidth pushes reversible triage and second-look actions into bounded autonomy. Higher bandwidth permits more human review.
Non-satellite positioning accuracy under degradation	Drift and confidence of terrain-relative, inertial, visual, or anchor-based positioning when satellite positioning is denied, shadowed, or spoofed.	Whether a candidate is actionable as a point, an area, or only a search cue.	Tens of metres can support asset commitment. Hundreds of metres turns a "find" into a re-search area and tightens the authority boundary.

MEASUREMENT	WHAT MUST BE MEASURED	DESIGN FORK IT DECIDES	EFFECT ON THE DESIGN
Connectivity across the search area	Uplink and downlink availability, latency, jitter, packet loss, and outage duration across terrain and weather.	Whether work collapses to drone and node or expands to regional edge and core.	Broken connectivity pushes work down. Reliable low-latency connectivity allows second-look inference, shared context, and coordination to move upward.
Mobile node capacity	Number of drones, streams, detections, claims, and operators a local node can support while disconnected.	Whether the node can serve as the local fleet brain during partition.	Higher capacity supports local fusion and triage. Lower capacity forces more filtering onto the drone and more conservative tasking.
Regional edge latency and availability	Round-trip latency, uptime, queue depth, and tail behavior from mobile nodes to the regional edge during an incident.	Whether the regional edge can sit in the live coordination loop.	Low, reliable latency supports federation and shared incident context. High or unstable latency pushes coordination back toward mobile nodes.
Confidential-inference overhead	Performance cost of attestation, encryption, protected execution, access checks, and audit logging for sensitive inference.	Whether sensitive matching or privacy-preserving analysis can run regionally.	Low overhead supports regional confidential inference. High overhead narrows the workload or pushes it to later core processing.
Cross-agency policy latency	Time to evaluate access, sharing, retention, and minimization rules across agencies.	Whether governance can stay in the live path.	Fast evaluation supports governed federation. Slow evaluation forces narrower sharing, pre-negotiated rules, or human escalation.
Evidence schema compatibility	Whether claims, provenance, confidence, coordinates, model versions, and policy labels stay parseable across nodes and agencies.	Whether claims can propagate safely across the system.	Stable schemas support federation. Schema mismatch turns coordination into manual reconciliation.
Trust-signal reliability	Accuracy of sensor-health models, anomaly checks, spoofing detection, and behavior-boundary enforcement.	Whether a claim can be trusted enough to propagate or trigger action.	Strong trust signals allow faster sharing and escalation. Weak signals require corroboration, quarantine, or human review.

MEASUREMENT	WHAT MUST BE MEASURED	DESIGN FORK IT DECIDES	EFFECT ON THE DESIGN
Event-log completeness under partition	Whether local logs capture claims, decisions, policy checks, and tasking events while disconnected, then reconcile cleanly.	Whether the system can preserve auditability under degraded connectivity.	Complete logs support offline operation and later reconciliation. Gaps reduce trust and may limit autonomy.
Federation conflict rate	Frequency of duplicate candidates, overlapping cells, inconsistent reports, and contested tasking across agencies.	Whether regional coordination is worth the cost.	High conflict rate strengthens the case for regional edge federation. Low conflict rate may leave more coordination node-local.

First measurements to run

Three measurements come first because they move the largest parts of the design.

1. **Sustained detector performance on the airframe, hot.** This decides whether the drone can run real perception or only produce cues for the mobile node. The relevant measurement is sustained frame-rate-at-accuracy after the full airframe system reaches operating temperature, not cold-bench performance.
2. **Non-satellite positioning drift under degradation.** This decides whether a candidate can become an actionable point. If degraded positioning holds to tens of metres, the system can support precise asset commitment. If it drifts to hundreds, the output becomes a re-search area and the authority boundary tightens.
3. **Operator cognitive bandwidth under load.** This decides how much evidence can be routed to a human before oversight becomes a rubber stamp. The system's autonomy boundary depends on the number and quality of decisions a real operator can make under fatigue, split attention, and time pressure.

How to use the backlog

The backlog is the test plan for the constraint map, not a list of caveats. Each measurement should produce one of three outcomes:

OUTCOME	MEANING
Constraint holds	The assumed placement, authority, or governance decision remains plausible.
Constraint fails	The function moves, degrades, or requires a different authority boundary.
Constraint is unstable	The system needs a runtime policy: degrade, shed, cache, buffer, quarantine, or escalate.

The implementation path is therefore to measure the constraints, then let the architecture move.

Appendix B: References and influences

The table below is a reading map for the areas of inquiry behind the essay. It points to public sources, adjacent research, operational doctrine, and technical analogues that informed the reasoning. It is illustrative rather than exhaustive. The references do not prove that the composed search-and-rescue system exists or that these pieces compose cleanly into a working whole. They show where the component ideas come from, where the analogues are strongest, and where the essay is carrying a design requirement rather than a settled practice.

AREA OF INQUIRY	USEFUL REFERENCES AND ANALOGUES
Operational AI as a distributed-systems problem (the lens)	Harnessing Edge AI to Strengthen National Security (CSIS, 2025); Distributed Mixture-of-Agents for Edge Inference (Mitra et al., 2024)
Cognition as a portfolio of bounded functions across tiers	Agentic AI Meets Edge Computing in Autonomous UAV Swarms (arXiv preprint, 2026); Edge Large AI Models: Collaborative Deployment and IoT Applications (HKUST, 2025); Distributed Mixture-of-Agents for Edge Inference (2024)
Four-layer placement (drone / node / regional edge / core)	Distributed Edge Inference Changes Everything (Akamai, 2025); From AI Factories to the Edge (Akamai / NVIDIA GTC, 2026); System Design for Heterogeneous GPU Workloads on NVIDIA Holoscan (NVIDIA, 2024)
Regional edge as a persistent, cross-incident layer (residency, cross-agency federation)	AWS European Sovereign Cloud (AWS, 2026, on data residency); Sovereign Control with Global Reach (Equinix, 2026); GFIPM cross-agency federation (DOJ BJA)
On-board detection, and the benchmark-vs-field gap under heat and motion	EdgeYOLO: An Edge-Real-Time Object Detector (BIT, 2023); Beyond Benchmarks: Continuous Edge Inference for Fine-Grained Roadside Perception (IISER Bhopal, 2026); Heterogeneous GPU Workloads on Holoscan (NVIDIA, 2024)
Model-size ladder: distillation and quantization to phone-class	Gemma open-weight on-device family (Google, 2026); MINIONS on-device / cloud collaboration (Stanford / Together AI, ICML 2025); OdysseyLLM W4A8 quantization (2023)
Bounded specialized models and the safety case	SAFE-AI: A Framework for Securing AI-Enabled Systems (MITRE); Verified ML Infrastructure: Formal Methods for Trustworthy AI Deployment (RAND); NIST AI RMF 1.0 (NIST, 2023)
Coordination under limited connectivity (multi-UAV)	AutoSOS: Multi-UAV Systems for Maritime SAR with Lightweight AI and Edge Computing (2020); Agentic AI Meets Edge Computing in UAV Swarms (arXiv preprint, 2026); A2A Protocol Specification (software-agent analogy for signed messaging, 2026)
Agentic cross-agency coordination and zero-trust across agents (the flood example)	Agentic AI Meets Edge Computing in Autonomous UAV Swarms (disaster-response template, 2026); Secure Multi-LLM Agentic AI and Agentification for Edge General Intelligence by Zero-Trust: A Survey (per-agent identity, continuous verification, audited exchange, 2025); Toward Edge General Intelligence with Multiple-LLM: Architecture, Trust, and Orchestration (orchestrate by constraint, 2025); Heterogeneous LLM Multi-Agent Systems (X-MAS) (different models matched to functions, 2025); GFIPM cross-agency federation (DOJ BJA)

AREA OF INQUIRY	USEFUL REFERENCES AND ANALOGUES
Trust plane: Zero Trust as evidence control; agent identity, continuous verification, and attested execution	Zero Trust for AI Agents (Anthropic, 2026); The Agentic Trust Framework (CSA, 2026); Applying Zero Trust to AI Agents (SANS, 2026); Confidential Inference Systems (Anthropic / Pattern Labs, 2025, on attestation); GFIPM federated identity (DOJ BJA)
Enforce-in-path: signed messages, validate-without-issuer, deterministic failsafe	A2A Protocol Specification (optional JWS signing; software-agent analogy, 2026); Securing Agentic AI: Threat Model and Mitigation (AWS, 2025); DoD Directive 3000.09: Autonomy in Weapon Systems (failsafe / guardrails, 2023)
Data governance, minimization, and confidential inference off the core	Confidential Inference Systems: Design Principles and Security Risks (Anthropic / Pattern Labs, 2025); Confidential LLM Inference: Performance and Cost Across CPU and GPU TEEs (ETH Zurich, 2025); Enhancing AI Inference Security with Confidential Computing (Red Hat / Tinfoil, 2025)
Evidence chain-of-custody and provenance (*partial grounding; treated as a requirement, not a settled standard*)	Digital Evidence and the U.S. Criminal Justice System (RAND / NIJ, 2015); GFIPM (audit and credentialing, DOJ BJA)
Positioning under GNSS denial or spoofing; drone-as-localization-source	SARDO: Automated SAR Drone-based Victim Localization (IEEE, 2020); AutoSOS multi-UAV maritime SAR (2020)
Bandwidth-aware drone video and task-oriented event extraction	Bandwidth-efficient Live Video Analytics for Drones via Edge Computing (CMU / Intel, 2018); Task-Oriented Communication for Edge Video Analytics (HKUST, 2024); Ekya: Continuous Learning of Video Analytics on the Edge (2020)
Autonomy ladder and the human boundary on irreversible action	EU AI Act, Article 14: Human Oversight (2024); NIST AI RMF 1.0 (2023); AI and the Future of Warfare (Cummings, Chatham House, 2017, on brittleness); Why Risk Should Determine Your AI Architecture (IBM, 2026)
Human judgment as a designed control, not a rubber stamp	Why Amazon Hates 'Human-in-the-Loop' AI Governance (The Register, on Brandwine / AWS, 2026); Why 'Human in the Loop' Alone Is Not a Governance Strategy (IBM, 2026); The Impact of AI Errors in a Human-in-the-Loop Process (Agudo et al., 2024); Keeping Humans 'in the Loop' (CIGI, 2020)
Civilian assurance base: response-robot and UAS test-and-evaluation	Performance of Emergency Response Robots (NIST); Aerial Drone Tests, Level 3-5 (NIST, 2024); DECISIVE sUAS Subterranean Test Methods (UMass Lowell / DEVCOM-SC, 2022)
First-responder UAS guidance and regulation (incl. Remote ID)	DHS / FEMA Guidance for UAS in Public Safety Missions (2024); Drone (UAS) Guidelines (GEMA, Remote ID); A Review of the Operational Use of UAS in Public Safety Incidents (CU Boulder, 2022)
SAR / US&R / ICS operational doctrine (the human coordination architecture)	Field Operations Guide ICS 420-1 (USFA / NFA, 2016); US&R Task Force NIMS resource typing (FEMA); Successful State SAR Program Design (NASAR, 2022)
Central retraining vs in-field learning; privacy-preserving on-device methods	Ekya: Continuous Learning on the Edge (2020); Communication-Efficient Learning from Decentralized Data (FedAvg) (McMahan et al., 2017); Private Federated Learning in Real-World Application (Apple, 2025)
Inference-serving throughput (the category error in Section 10)	PagedAttention / vLLM (SOSP '23); Orca: Iteration-Level Scheduling (OSDI '22); Sarathi: Chunked Prefills (2023); Mastering LLM Inference Optimization (NVIDIA, 2023)
Day-2 fleet ops: rolling deploy, rollback, autoscale, scale-to-zero	Rolling Deployments for Zero-Downtime Model Updates (Baseten, 2026); Routing for Serverless Servers (Modal, 2026); Operationalize Day-2 Services (Akamai, 2025)

AREA OF INQUIRY	USEFUL REFERENCES AND ANALOGUES
Inference-serving substrate (serving, routing, scaling, observability, rollback of bounded models)	DistServe: prefill/decode disaggregation for goodput (OSDI '24); Sarathi-Serve: chunked-prefill stall-free scheduling (OSDI '24); Mooncake: KVCache-centric disaggregated serving (FAST '25); open-source serving stacks, KV-cache-aware routing, and inference gateways are treated as examples of the pattern, not as pieces the essay depends on
Agent-operating substrate (agent identity, governed tool/ API gateways, claim exchange, policy enforcement, audit)	W3C Verifiable Credentials Use Cases; W3C Agent Identity Registry Protocol CG (verifiable agent identity, 2026); OAuth 2.0 Authorization Server Metadata (RFC 8414) and agent-authorization extensions; Governance Gaps in Agent Interoperability Protocols (what today's protocols cannot yet express, 2026); content provenance and transparency (C2PA, SCITT / RFC 9943)

A few elements are deliberately carried as reasoning aids rather than findings. The four-tier substrate is a framing choice. The placement table and degradation policy are illustrative, not tuned configurations. Request-signing for aircraft telemetry and tasking is treated as a design assumption: the pattern is established in software systems and agent protocols, but this essay does not establish it as standard practice for this medium. In-field continuous learning for a safety-relevant perception function remains speculative; the working position here is central validation before live perception changes.

One example is evidence chain-of-custody for autonomous-sensor evidence. Civilian digital-evidence guidance exists, and public-safety identity and audit frameworks exist, but a mature civilian standard for autonomous-sensor custody is not treated here as settled. In this essay, that becomes a design requirement to reason from, not a finding.