ARCHITECTURE

System of Systems: An Architecture Overview for Regional Wildfire Defense

Most regions already own the pieces of a modern wildfire defense — cameras, sensors, aircraft, dispatch, an incident command structure. What they don't have is the layer that makes those pieces act as one system, with a human in command of every consequential decision. This paper describes how Inari Watch builds that layer.

The architecture in brief

A region exposed to wildfire rarely lacks for technology. It lacks integration. The camera network, the sensor feeds, the aircraft, the dispatch console, and the incident command structure each work — but they work as silos, handing the problem to a human at every seam. That hand-off is where minutes and coordination are lost; we treat it at length in *The Integration Gap*.

Inari Watch closes it by building a system of systems. Within a defined **protected region** — a county, a utility service territory, or a federal installation, each with a clear boundary and an identifiable risk owner — we unify the sensing, the aerial assets, and the ground control into one coordinated capability. The architecture has two parts: **three physical layers**, and the **agentic coordination fabric** that unifies them. The physical layers are largely built from proven components. The novelty, and the safety, live in the fabric.

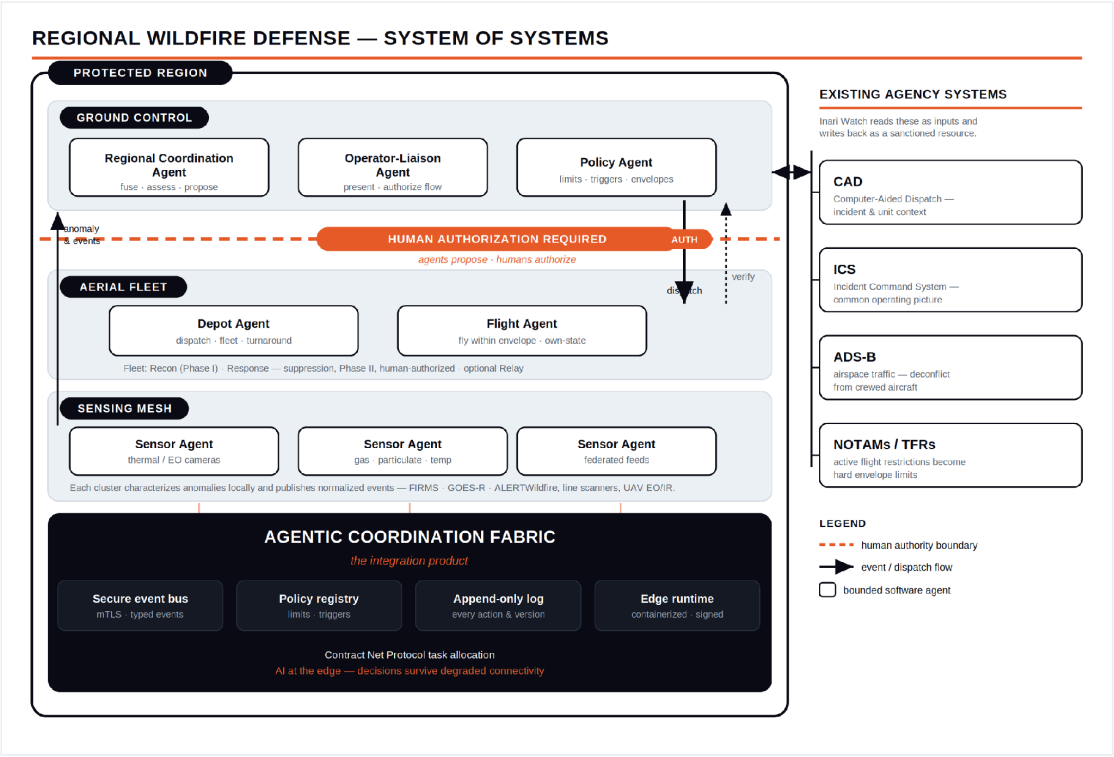

Figure 1 — The three physical layers, the coordination fabric that unifies them, the human authority boundary that governs every kinetic action, and the existing agency systems the platform integrates with.

The three physical layers

Sensing mesh. The sensing layer is heterogeneous by design. Each modality — thermal and optical, chemical, satellite — has different strengths, blind spots, and confidence characteristics, and treating them as a single mesh rather than parallel silos is the first product of integration. The mesh draws on fixed thermal and optical cameras on towers, ridgelines, and infrastructure; gas, particulate, and temperature sensors along corridors and at canyon outlets; line scanners and IR arrays in the highest-risk bands such as transmission corridors and the wildland-urban interface; federated remote-sensing feeds including NASA FIRMS (VIIRS and MODIS), NOAA GOES-R, JPSS, and ALERTWildfire; and EO/IR imagery from UAVs on patrol. Each sensor cluster carries an edge compute node running a **sensor agent**, which characterizes anomalies locally, applies false-positive filters, and publishes a normalized event to the region — rather than streaming raw data upstream. The region receives a common vocabulary of events, not a flood of incompatible feeds.

Aerial fleet. The UAVs are organized as a regional fleet in defined classes. **Recon UAVs**, carrying EO/IR sensors for patrol routes and verification dispatch, are the primary aerial asset in Phase I. **Response UAVs**, carrying the precision suppressant payload, are a Phase II capability for human-authorized suppression missions — they are not deployed in Phase I. Optional **relay UAVs** extend communications and atmospheric profiling into terrain shadows. Each aircraft runs a **flight agent** that maintains awareness of its own state — position, fuel, payload, health — accepts missions from its depot within a pre-approved envelope, and escalates anything outside that envelope for human authorization. In Phase I, the aircraft verify and patrol; they do not release anything.

Ground control. The ground layer is the physical and organizational backbone. Distributed **depots** near high-risk zones house the UAVs, launch and recovery infrastructure, and an edge cluster running the **depot agent**. A **regional operations console** is where operators supervise the system, approve elevated-authority actions, and intervene at will; it hosts the regional coordination agent and the operator-liaison agents. And the ground layer is where the platform meets the systems a region already runs — dispatch, incident command, airspace — which we cover below.

The coordination fabric

The fabric is the integration product — the part that is genuinely new. It has four components. A **secure event bus** (mutually authenticated, signed messages) carries typed, versioned events between agents, so every component speaks the same language and nothing moves anonymously. A **shared policy registry** defines each agent class's authority limits, escalation triggers, and operational envelopes — the rules live in data, not buried in code. An **append-only decision log** captures every agent action against the policy version in effect, so the system is reconstructable end to end. And an **edge runtime** hosts the agents as signed containers on compute nodes throughout the region, with admission controls, so decisions are

made locally and survive the degraded connectivity a wildfire produces. The cloud supports the system; it is never a single point of failure.

Tasks are allocated across the fabric using the **Contract Net Protocol**: when work needs doing — a verification flight, a cross-depot reallocation — agents bid within their policy limits and the region awards. It is a well-understood distributed pattern, chosen because it degrades gracefully and keeps allocation explainable.

The six agents — and where each one's authority stops

The fabric is operated by six classes of bounded software agent. Each has a defined scope and, more importantly, a hard limit on what it may do on its own.

Agent	Lives in	What it does	Where its authority stops
Sensor Agent (per cluster)	Sensing mesh	Characterizes anomalies locally, filters false positives, publishes normalized events	Reports only — cannot task aircraft or commit any resource
Flight Agent (per UAV)	Aerial fleet	Flies approved missions, tracks its own state, runs onboard safety behaviors	No payload release and no flight outside the pre-approved envelope without human authorization
Depot Agent (per depot)	Aerial fleet	Proposes and dispatches verification missions within envelope; manages fleet, charging, turnaround	Cannot authorize a payload-equipped or out-of-envelope mission on its own
Regional Coordination Agent (per region)	Ground control	Fuses sensor events, forms incident hypotheses, computes risk, proposes responses, allocates resources	Routes every kinetic action and out-of-envelope dispatch to a human; cannot change regional policy
Operator-Liaison Agent (per operator)	Ground control	Presents prioritized, explained, ready-to-authorize decisions; manages authorization workflows and veto windows	Cannot authorize anything on the operator's behalf
Policy Agent	Ground control	Holds and enforces the policy registry — authority limits, escalation triggers, operational envelopes	Defines the boundaries; does not act inside them, and policy changes are themselves audited

The pattern across all six is the same, and it is the most important thing in the architecture: **agents propose; humans authorize.** Software perceives, correlates, recommends, and executes inside an approved envelope. Any action that releases a payload, commits a regional resource beyond a set threshold, or steps outside an envelope requires explicit, contemporaneous human authorization — enforced in the policy layer, not left to convention. So when someone asks whether the aircraft are autonomous, the precise answer is that the system is autonomous in how it senses and coordinates, and never in the decision to act.

End to end: a representative ignition

The clearest way to understand the architecture is to follow one event through it. A new ignition flows like this:

1. **Detect.** A thermal camera's sensor agent registers an anomaly, characterizes it locally, filters obvious false positives, and publishes a typed event to the region.
2. **Fuse.** The regional coordination agent correlates that event against other sensor inputs and recent history, forms an incident hypothesis, and assigns a confidence level.
3. **Assess.** It computes a risk profile — location, fuel, weather, values at risk — and determines whether the incident warrants a response.
4. **Propose.** If it does, the agent proposes one: which asset, from which depot, on what route, allocated across the fleet within pre-approved policy.
5. **Present.** The proposal rises through the operator-liaison agent to a human at the regional console — as a prioritized, explained, ready-to-authorize decision, not a raw alert.
6. **Authorize.** The operator approves, modifies, or declines. Nothing kinetic happens without this step.
7. **Dispatch.** On approval, the depot agent tasks a recon UAV; its flight agent flies the mission within the approved envelope and escalates anything outside it.
8. **Verify and log.** The aircraft returns verification imagery, which feeds back into the coordination agent's incident record. Every decision along the path is logged against the policy version in effect.

In Phase I, this flow ends at verification — confirm the fire, characterize it, and hand a clear picture to the people who respond. The suppression branch, in which a human authorizes a Response UAV, is a Phase II capability built on exactly this authorization spine. The spine does not change; a capability is added to it.

Integration with existing agency systems

A wildfire program does not need another standalone tool. It needs a new capability that fits the systems it already runs. Inari Watch is designed to operate as a sanctioned resource inside those systems — reading them as inputs and writing back to them — never as a parallel authority. Four interfaces matter most.

CAD — Computer-Aided Dispatch. The platform reads incident and unit context from dispatch, and writes confirmed detections back into it, so they surface in the dispatcher's existing console rather than on a separate screen no one is watching.

ICS — Incident Command System. The platform aligns to the NIMS/ICS structure, contributes to the common operating picture, and produces ICS-compatible status products. When a fire is under incident command, the system answers to that command like any other resource.

ADS-B. The platform ingests ADS-B traffic to maintain real-time airspace awareness and keep its UAVs deconflicted from the crewed aircraft — air tankers, helicopters, spotters — that share wildfire airspace. Deconfliction is treated as a safety-critical function, not a convenience.

NOTAMs and TFRs. The platform reads active Notices to Air Missions and Temporary Flight Restrictions, and an active wildfire TFR becomes a hard constraint inside the flight and depot agents' envelopes. The system does not fly where it has not been cleared to fly.

For a utility or agency wildfire program, this is the part that determines whether the system is adoptable at all. It fits dispatch, it answers to incident command, it deconflicts the airspace, it respects flight restrictions, and it logs every decision for after-action and liability review. Those are the questions a program manager has to answer before anything else — and the architecture is built to answer them yes.

What this means for an evaluation

This is a deliberately unglamorous architecture. Its value is not a single breakthrough device; it is coordination — making proven components function as one system, under human command, inside the structures a region already operates. If your region already owns most of the pieces, the useful next step is a short discovery call to map how they connect in your specific operational picture: your sensors, your airspace, your dispatch and command structure, and the authority boundaries that matter to you.

About Inari Watch

Inari Watch is a veteran-founded company building an integrated system of systems for regional wildfire defense. We unify the sensors, aerial assets, and data feeds a region already depends on into one coordinated capability — connected by AI at the edge and bounded agentic coordination, with humans in command of every consequential decision.