Nick Clark · Published April 25, 2026 · PDF

How to Build a Shared Spatial Map That Multiple Robots Can Trust

If you run a fleet of robots, vehicles, or drones, sooner or later they need one shared picture of where things are, and every one of them has to be able to tell which parts of that picture are worth acting on. This guide walks through an architectural approach to that problem: make each observation carry its own authority credential, identity proof, and provenance, and let every consumer decide admissibility for itself. The approach described here is disclosed in U.S. Provisional Application No. 64/049,409 (it is an architecture you build, not a shipping library). The home inventive step is the Governed Spatial Mesh inventive step.

What You Are Building

You have more than one autonomous machine operating in the same space: warehouse robots on a shared floor, vehicles on a corridor, drones over a site, vessels in a port. Each one senses part of the world. You want them to pool what they see into a single shared spatial map, so a robot can act on a hazard it never saw directly because a neighbor saw it first.

The hard part is not merging point clouds. The hard part is trust. The moment a robot acts on an observation it did not produce itself, you have to answer: who said this, are they allowed to say it, is it really them, and can I reconstruct why I believed it after

something goes wrong? A shared map that any participant can silently poison is worse than no shared map, because it launders a bad input into a fleet-wide decision.

This guide describes an architecture for a shared spatial map in which trust is a property of every individual observation, not of the network as a whole. It is the approach disclosed in U.S. Provisional Application No. 64/049,409. You will still write the code yourself; what follows is the design.

Why the Obvious Approaches Fall Short

The common approaches each solve part of the problem and leave a structural gap.

Per-robot perception with no sharing. Every robot builds its own world model from its own sensors. This is robust and simple, and it is what most fleets do today. But each robot only knows what it can see from where it is, it cannot see around a blind corner or behind an occlusion, and every robot pays to detect the same object independently. There is no shared picture at all.

A central map server. Push every robot's detections to one cloud service that maintains the canonical map and serves it back. This gives you a shared picture, but it makes the server a single point of failure and a single point of compromise, it requires connectivity the robots may not have, and it typically resolves conflicts by last-writer-wins or a fixed fusion rule. Nothing in a raw detection tells the server, or a downstream consumer, whether the sender was authorized to make that claim.

Signed messages over a mesh. Have robots gossip detections peer to peer and sign each message. This is a real improvement and removes the central dependency. But an ordinary signature carries binary semantics: the signature verifies or it does not. It does not tell a consumer that a municipal traffic device outranks a random aftermarket sensor, and static keys are vulnerable to theft and replay, which is exactly the failure

mode you cannot tolerate in a system that drives actuators. Public-key infrastructure also assumes enrollment with a certificate authority before a device can be trusted, which is awkward for devices deployed into environments with no connectivity.

The gap in all three is the same: the observation itself does not carry graded, machine-consumable trust, and identity is either absent or a static secret. The architecture below closes that gap.

The Architecture

The disclosure describes an architectural inversion: instead of each unit independently building a world model from its own sensors, the environment and its participants distribute *governed observations*, and each operating unit consumes them into its own shared spatial picture. Four elements make those observations trustworthy.

1. The governed observation is the unit of exchange. Every communication in the mesh, from any contributor, is formatted as a governed observation with a fixed set of fields: an authority-credential field, a dynamic-device-hash field for identity, a spatial-reference field, a temporal-reference field, a time-to-live field, a domain-specific payload, and a lineage field recording provenance with a cryptographic integrity attestation. The disclosure specifies a concrete byte layout for these fields. The key design decision is uniformity: a passive marker, a fixed sentinel, a coordinating agent, and a moving robot all emit the same primitive, so one admission path handles everything.

2. Authority is a credential with hierarchical semantics. Each observation carries an authority credential that encodes an issuing-authority identifier, a scope, a temporal-validity window, a device-binding attestation, and a cryptographic attestation (the disclosure does not fix the primitive; digital signatures, threshold signatures, zero-knowledge, or post-quantum attestations are all in scope). Consumers evaluate that credential against a governance-configurable *authority taxonomy*: a hierarchy where

each level maps to a behavioral response, an evidential weight, a rule for whether observations at that level may be injected into the planner, and a supersession rule for conflicts with lower levels. The disclosure gives domain examples (a roadway taxonomy from regulatory down to advisory and no-authority; defense, healthcare, and warehouse taxonomies). This is the difference from a plain signature: trust is graded, not binary, and the grading is consumed by the robot's decision logic.

3. Identity is continuity, not a static key. Each device computes a *dynamic device hash* from device-specific entropy, sensor readings, configuration, clock state, and the content of prior transmissions, and attaches it to every message. The hash is designed to evolve gradually across successive transmissions, tracking the device's real operational state. Receivers keep a history of hashes per sender and run a *trust-slope validator* that checks whether a newly received hash is consistent with genuine evolution. A spoofer that steals a static credential still cannot reproduce the continuous evolution of the genuine device's hash sequence, so a discontinuity flags spoofing or replay even when the credential itself checks out. This approach needs no enrollment with a central certificate authority before deployment, stores no long-lived secret on the device, and tolerates maintenance and configuration changes through a governance-policy-defined tolerance window.

4. Every consumer runs a composite admissibility evaluator. No observation is trusted because it arrived. Each robot runs an evaluator that ingests governed observations, computes an effective evidential weight from multiple factors (authority level, staleness, sensing-modality reliability, dispositional context, reputation track record, integrity, and the continuity-validation output from the trust-slope validator), and produces one of six outcomes: admit, gate (admit at reduced weight or with constraints), defer (hold pending corroboration), solicit (actively query for corroborating observations), reject (with a reason class such as insufficient authority, failed continuity validation, stale, or failed corroboration), or escalate. Crucially, it emits a governed admissibility-determination observation recording the inputs, the weights, the factors, and the outcome, so every decision is reconstructable.

These four elements compose into the shared map itself, which the disclosure calls a shared environmental world view. It admits observations through the evaluator, resolves conflicts by *evidentially-weighted aggregation* rather than last-writer-wins, keeps governance-chain lineage for every admission and consumption, supports governance-credentialed retraction and correction (a retracted observation stays in the chain as retracted-and-superseded, never deleted, preserving audit), and can reconstruct the view state at any prior time for forensics. Robots can even build a shared coordinate frame cooperatively through inter-agent ranging with adversarial-range rejection, so the map does not depend on any single external positioning source.

How to Approach the Build

You implement this yourself. A workable order:

Step 1: Define the observation primitive. Fix a serialization for the seven fields (authority credential, device hash, spatial reference, temporal reference, time-to-live, payload, lineage). Make it uniform across every device class from the start; the whole architecture depends on one admission path.

An illustrative interface sketch, faithful to the disclosed fields (not shippable code):

```
GovernedObservation {
  authority_credential   // issuer id, scope, validity, device binding, atte
  dynamic_device_hash    // identity-continuity token
  spatial_reference      // geographic, mesh-derived, or local frame
  temporal_reference     // global, mesh-derived, or local clock
  time_to_live           // validity duration
  payload                // domain-specific content
  lineage                // contributing device, source refs, derivation, in
}
```

Step 2: Choose the attestation and the authority taxonomy. Pick a cryptographic attestation mechanism, then write down your taxonomy: the trust levels for your domain, and for each level the behavioral response, evidential weight, planner-injection rule, and supersession rule. This is governance policy, not code you bake in; keep it updatable.

Step 3: Implement continuity identity. Build the dynamic-device-hash generator (mix in per-device entropy, sensor and config state, clock, and prior-transmission content), the per-sender hash history store on receivers, and the trust-slope validator with a configurable tolerance window. Test it against the spoofing case: a device that presents a valid credential but cannot reproduce a genuine hash evolution should fail continuity validation.

Step 4: Build the composite admissibility evaluator. Implement the multi-factor weight computer and the six outcomes. Do not collapse it to admit-or-reject; the defer and solicit outcomes are what let a robot wait for corroboration instead of acting on a lone unverified claim. Emit the admissibility-determination record every time.

Step 5: Assemble the shared world view. Wire admission through the evaluator, use evidentially-weighted aggregation for conflicts, record lineage on every event, and implement retraction as supersession rather than deletion.

Step 6: Add graceful degradation. The disclosure describes progressive enhancement: a unit operates on its own sensors and peer observations when no infrastructure is present, and gains confidence and capability as more credentialed devices appear. Have your confidence governor lower execution readiness when credentialed device density drops at runtime, and restore it when the unit re-enters a richer region. Build this in rather than assuming full coverage.

Tradeoffs to plan for: every message carries credential, hash, and lineage overhead, so budget bandwidth and storage; the trust-slope tolerance window is a real tuning problem (too tight rejects legitimate maintenance events, too loose lets spoofers

through); and evidentially-weighted aggregation needs enough independent corroborating sources to be meaningful, which is why the solicit outcome exists.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and no SDK behind this guide. Every component (the observation serialization, the attestation choice, the taxonomy, the hash generator, the trust-slope validator, the evaluator, the aggregation logic) is something you design and implement for your domain, and the disclosure describes them as architectural primitives rather than tuned implementations.

It is a patent-disclosed approach, not a benchmarked or productized system. The disclosure does not state accuracy figures, latency numbers, or throughput, and neither does this guide. Any performance you get depends entirely on your ranging modalities, device density, cryptographic choices, and tuning, and you should measure your own build rather than assume the architecture guarantees a result. The trust properties described here are structural (graded authority, continuity identity, per-consumer admissibility, reconstructable lineage); they are only as strong as the cryptographic attestation you choose and the governance policy you write. If your fleet is a single robot, or all robots fully trust one central authority you already control end to end, the machinery here is more than you need.

Disclosure Scope

The architecture described in this guide, including the governed observation primitive, the authority taxonomy, the continuity-based (trust-slope) device identity mechanism, the composite admissibility evaluator, and the shared environmental world view with evidentially-weighted aggregation, is disclosed in U.S. Provisional Application No. 64/049,409. This guide is educational. It explains an architectural approach a

developer can implement independently; it is not a warranty, a specification of a released product, or an offer of software, and nothing here should be read as a performance guarantee.

Governed Spatial Mesh (/spatial-mesh)

All 40 steps → (/inventive-steps)

The environment holds perception, not the unit. Every transmission carries authority.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Governed Spatial Mesh: The Architecture Where the Environment Holds Perception (/articles/governed-spatial-mesh-the-architecture-where-the-environment-holds-perception)

SECONDARY TECHNICAL

- Architectural Inversion: Data Carries Authority (/articles/spatial-mesh/architectural-inversion)
- Three-Tier Environmental Device Architecture (/articles/spatial-mesh/three-tier-devices)
- Governed Observation: Authority-Credentialed Bytes on the Wire (/articles/spatial-mesh/governed-observation)
- Authority Taxonomy: Hierarchical Trust Structure for Governed Observations (/articles/spatial-mesh/authority-taxonomy)
- Marker Stored-Data Byte Layout (/articles/spatial-mesh/marker-byte-layout)
- Governed Mesh Message Format: Medium-Agnostic Message Structure (/articles/spatial-mesh/mesh-wire-format)
- Dynamic Device Hash for Continuity (/articles/spatial-mesh/dynamic-device-hash)
- Hop-History Relay (/articles/spatial-mesh/hop-history-relay)
- Rateless FEC for Lossy Mesh Media (/articles/spatial-mesh/rateless-fec)
- Mobile Store-and-Forward (/articles/spatial-mesh/mobile-store-and-forward)
- Firmware Updates Through the Mesh (/articles/spatial-mesh/firmware-via-mesh)
- Governance Policy Distribution Through the Mesh (/articles/spatial-mesh/policy-via-mesh)
- The World Broadcasts Authority: Navigation as the Physical Dual of Semantic Discovery (/articles/spatial-mesh/the-world-broadcasts-authority)

APPLICATIONS · GENERAL

- Coalition JADC2 Without a Single Data Owner: A Governed Spatial Mesh for Contested Battlespace (/articles/spatial-mesh/defense-battlespace-mesh)
- Cross-Organizational Industrial Digital Twins Without Platform Lock-In: A Governed Spatial Mesh Architecture (/articles/spatial-mesh/industrial-digital-twin-mesh)
- Spoof-Resistant Ship Tracking and Cross-Flag Port Coordination: A Governed Spatial Mesh for Maritime Operations (/articles/spatial-mesh/maritime-operations-mesh)
- Smart-City Sensor Mesh Without a Centralized Data Fabric: A Governed Spatial Mesh Approach (/articles/spatial-mesh/smart-city-spatial-mesh)
- Cross-Vendor Border and Perimeter Surveillance: A Governed Spatial Mesh Deployment (/articles/spatial-mesh/border-perimeter-mesh-deployment)
- EU AI Act Compliance for High-Risk Spatial Autonomy Systems (/articles/spatial-mesh/eu-ai-act-spatial-compliance)
- Pharmaceutical Cold-Chain Traceability: Unified Custody and Temperature Lineage for DSCSA and GDP Compliance (/articles/spatial-mesh/pharmaceutical-cold-chain-mesh)
- Rural Broadband Mesh Alternative for Last-Mile Connectivity (/articles/spatial-mesh/rural-mesh-broadband-substitute)
- Disaster Response Communications When Cellular Networks Fail: A Governed Spatial Mesh Deployment (/articles/spatial-mesh/scenario-disaster-deployment)

APPLICATIONS · SPECIFIC

- Anduril Lattice Alternative: Cross-Authority Mesh Substrate for Coalition Autonomy (/articles/spatial-mesh/anduril-lattice)
- AWS GovCloud Alternative for Federated Defense: Governed Spatial Mesh (/articles/spatial-mesh/aws-govcloud-defense)
- Palantir Gotham vs Governed Spatial Mesh: Cross-Authority Data Sharing (/articles/spatial-mesh/palantir-gotham)
- Cisco Hypershield vs Governed Cross-Authority Security Mesh (/articles/spatial-mesh/cisco-hypershield)
- Esri ArcGIS vs Governed Spatial Mesh: Cross-Authority Composition (/articles/spatial-mesh/esri-geospatial-platform)
- Lockheed Martin JADC2 vs a Governed Cross-Service Mesh (/articles/spatial-mesh/lockheed-jadc2)
- Governed Spatial Mesh Beyond Northrop ABMS and JADC2 (/articles/spatial-mesh/northrop-jadc2-abms)
- Raytheon RTX Defense Mesh: Governed Spatial Mesh vs Program-by-Program Integration (/articles/spatial-mesh/raytheon-rtx-defense-mesh)

- DIMO Network vs Governed Spatial Mesh: Credentialed Vehicle Observations (/articles/spatial-mesh/dimo-network)
- Helium Network vs Governed Spatial Mesh: DePIN Coverage Attestation (/articles/spatial-mesh/helium-network)
- Hivemapper Alternative: Governed Spatial Mesh for Decentralized Mapping (/articles/spatial-mesh/hivemapper-mapping)
- BAE Systems Defense Programs vs a Governed Spatial Mesh (/articles/spatial-mesh/bae-systems-defense-mesh)
- Governed Spatial Mesh vs Booz Allen Hamilton JADC2 Integration (/articles/spatial-mesh/booz-allen-defense)
- CACI Defense Programs vs a Governed Spatial Mesh Substrate (/articles/spatial-mesh/caci-defense)
- General Dynamics Defense Programs vs a Governed Spatial Mesh (/articles/spatial-mesh/general-dynamics-defense)
- L3Harris Tactical Radios vs a Governed Cross-Vendor Spatial Mesh (/articles/spatial-mesh/l3harris-defense)
- Leidos Defense Programs vs a Governed Spatial Mesh Substrate (/articles/spatial-mesh/leidos-defense)
- Leonardo Tactical Mesh vs a Governed Spatial Mesh: Coalition PNT Beyond GNSS (/articles/spatial-mesh/leonardo-defense-mesh)
- MBDA Missile Systems vs a Governed Spatial Mesh for Coalition Kill Chains (/articles/spatial-mesh/mbda-missile-systems)
- Rheinmetall vs a Governed Coalition Spatial Substrate (/articles/spatial-mesh/rheinmetall-defense)
- SAIC Defense Programs vs a Governed Spatial Mesh Substrate (/articles/spatial-mesh/saic-defense)
- Thales Defense Mesh Alternative: Governed Spatial Mesh Beyond Link 16 and SYNAPS (/articles/spatial-mesh/thales-defense-mesh)
- Mobilicom Alternative: Governed Cross-Vendor Spatial Mesh for Tactical Drones (/articles/spatial-mesh/mobilicom-defense-comms)

Governed Spatial Mesh overview → (/spatial-mesh)