PolyCap — A Functional Overview for Infrastructure Architects

Affectron, LLC · v0.2 · September 2026 · For ideation, not implementation

What this document is

A map of the functionality PolyCap provides, written for people who design infrastructure and want to know what they could build on it. It stays at the altitude of *properties and capabilities*; wire formats, mechanisms, and conformance details live in the design set, which is available to partners under the conformance program. If a paragraph here makes you think "we could use that for...," it has done its job.

What PolyCap is, in one paragraph

PolyCap is an **authority and coordination substrate** for networks of people, devices, agents, and services. It answers one question everywhere, uniformly, with proof: *who may do what, to which thing, until when, and who can verify it afterward.* It rides on ordinary MQTT 5 — any standard broker, or no broker at all — and adds delegable, attenuable, expiring, revocable **capabilities**; a priority-disciplined **messaging fabric**; content-addressed **state and sync**; governed groups called **Venues**; **discovery** without surveillance; a bounded, non-speculative **economic layer** for infrastructure duties and metered services; and an **emergency lifeline** that works with the network switched off. It is *not* a blockchain, not a currency, not an identity provider, and not a replacement for your broker.

The mental model (three rules that explain most of the design)

Keys instead of accounts. Every participant is a keypair. There is no central account store and no login. Authority is held as signed capabilities; identity is optionally aliased for anonymity. Consequence for you: no identity provider to stand up, no session database to protect, and no "the auth service is down" failure mode.

Authority lives in the message, never in the connection. Transport authentication (TLS, broker credentials) proves you can *reach* a node; it grants no PolyCap authority whatsoever. Every frame carries its own proof. Consequence: PolyCap is transport-agnostic and broker-agnostic by construction — a malicious broker or bearer can drop, delay, or replay traffic, but it cannot forge or escalate authority.

Receipts instead of surveillance. Every authorized action can emit a signed, attributable, tamper-evident receipt naming who acted, under what authority, derived from whom, expiring when. Audit is a property of the wire, not a log someone remembered to keep. Consequence: regulated and safety-critical environments get attributable records without a central observer.

Functional inventory

Authority

Capabilities are **delegable** (hand a narrower one to another party), **attenuable** (the invariant `authority(out) ⊆ authority(in)` is enforced at every hand-off — no delegate can hold more than its grantor), **expiring**, and **revocable**, with revocation propagating through the network in seconds. Denials are **typed**: a refused action names precisely which authority was missing, never a silent failure. Exclusive roles (one settlement facilitator per venue, one operator per instrument) are held as **leases** backed by consensus, so "who is in charge" is never ambiguous. *Enables:* least-privilege agents and devices; delegation chains with a paper trail; safe hand-offs to automation.

Identity, aliases, and custody

Keypair identities with **alias strata** for anonymity (pseudonymous participation with rotation and per-context aliases); **custody tiers** from software keys to hardware-anchored keys (TrustZone-class secure worlds and secure elements are the design target); **attestation** that distinguishes claimed from hardware-attested properties, and weights trust accordingly. *Enables:* privacy-preserving participation; devices whose keys never leave silicon; honest trust scoring.

Messaging fabric

MQTT 5 native — the wire unit is a standard PUBLISH whose payload is a PolyCap frame. Traffic is classified into a closed set of **lanes** (object, event, sync, query, real-time signaling, telemetry, emergency), each mapped to one of four **priority bands** with strict scheduling: an emergency frame is never the victim of eviction while lower-priority traffic exists. Bearers include TCP, QUIC (where each priority band gets its own independently-recovered stream), WebSocket, BLE, and HTTP gateways. Nodes speak MQTT in three

roles: **client to any unmodified broker**, **embedded micro-broker** serving local devices, and **peer bridge** carrying the same packet stream node-to-node. *Enables:* drop-in adoption on existing broker estates; safety traffic that survives congestion; devices that keep working across bearer failover.

Mesh and routing

A gossip-family pub/sub mesh (per-topic, scored, churn-tolerant) plus a Kademlia-style discovery layer. Meshes are tuned per traffic family; the emergency lane does not gamble on probabilistic delivery — it floods its authorized scope. Locality and cost enter routing preference, so the topology concentrates onto nearby, reliable, inexpensive paths (CDN-like formation) while diversity floors keep it partition-resistant. Nodes can **island** — full local function when the wider network is unreachable — and heal with revocations processed first. *Enables:* planet-scale fan-out without a phone melting; regional efficiency; resilience to correlated outages.

State and sync

Content-addressed objects; range-based **reconciliation** that converges two replicas by exchanging fingerprints rather than full state; retained values that make the fabric behave as **shared memory implemented over message passing**; deduplication that spans reconnects. Offline-first by nature: a node that was away catches up by reconciling, not by replaying a log it might have missed. *Enables:* distributed digital twins; fleet state that survives disconnection; multi-writer world models with per-branch write authority.

Venues (governed groups)

A Venue is a scope with membership-as-capability, roles, policy overlays that can only *tighten* what the substrate allows, treasury and escrow, and optional consensus-backed leases. Nine closed templates cover the common shapes: **Family, Incident, Settlement, Boundary (external-system adapter), Marketplace, Compliance, Real-Time, Discovery, and Emergency**. Governance is *provenance, not types*: what a venue may do is determined by what authority it was granted, not by a special-case protocol. *Enables:* teams, fleets, incident rooms, marketplaces, and compliance domains built from the same parts; institutions layered above the substrate without the substrate granting them thrones.

Discovery and matching

Capabilities and competences are advertised as compact semantic vectors into named affinity spaces. Neighborhoods become visible only as **k-anonymous centroids** (no centroid exists below eight members; coverage is reported in buckets, never exact counts). Queries pass three gates — authority, disclosure class, and coverage attestation — and an unauthorized query learns nothing, not even that a neighborhood exists. "Guilds" are these emergent neighborhoods; formal guilds are Venues built above them. *Enables:* finding a

provider of a service without a directory anyone can scrape; expertise markets with privacy floors.

Economics (bounded by design)

LP is an internal, non-convertible unit with no off-ramp. Ordinary mesh participation is an unpaid commons; standing infrastructure **duties** (relay, storage, uplink, indexing, voting) are opt-in, staked, and rewarded from **bounded per-epoch budgets** (farming dilutes itself by arithmetic); consumed **services** are metered with witnessed evidence and fail closed without it. Escrow and settlement support leases, hosted execution, and device accounting. *Enables:* provisioning infrastructure without speculation; pay-per-use device and service accounting (industrial and lab equipment included); markets that cannot mint.

Real-time

Signaling for WebRTC-class sessions rides its own low-latency lane with deterministic glare resolution; media travels off-lane on negotiated paths. **Presence** is ephemeral, level-of-detail gated, and alias-disciplined — you receive presence at the resolution your relationship warrants. *Enables:* calls, collaboration, and spatial presence with the same authority model as everything else.

Emergency lifeline (EOB)

An offline bundle that keeps working when infrastructure fails: an **anonymous, unlinkable beacon** (a rotating alias that reveals nothing until a holder chooses to), tiered disclosure to responders with defined **credential classes**, drill mode, and structural silence when dormant. Emergency traffic occupies the highest priority band and carries no price. *Enables:* rescue and mass-casualty coordination that does not depend on the network being up or anyone being trusted in advance.

Executable artifacts and plugins

Hosted code receives authority only by **sealed capability injection** — no ambient access, no dynamic code loading, host-mediated spending. Native plugins pass a four-gate rule. The same discipline applies to device drivers and agent-authored scripts. *Enables:* running third-party or AI-generated automation against real equipment with bounded blast radius and a supply-chain provenance trail.

Boundary adapters

External systems attach through adapters that preserve the attenuation invariant at the seam: a telecom/6G boundary adapter, a spatial-web adapter for shared scene graphs (multi-writer twins with per-branch write handles), and a **steward** pattern for plain MQTT or vendor devices that cannot speak PolyCap themselves (a node holds authority on the device's behalf and emits receipts for its actions). The same pattern fronts agent-hardware standards: PolyCap sits at the actuation surface as a *reference monitor with receipts*.

Enables: governing legacy devices, industrial estates, and agent-operated instruments without modifying them.

Observability and configuration

Telemetry rides the lowest-priority lane with per-tier sampling and mandatory redaction classes (there is deliberately no unified device-health broadcast — health is inferred from behavior and scoped to consumers). Configuration is layered; **floors** (battery, thermal, data caps, retention, revocation coupling — 154 registered controls) are immutable per signed release and cannot be breached by any overlay or runtime governor. Corrupted configuration drops to a signed safe mode. *Enables:* fleets that cannot be misconfigured into unsafety; audit trails that are also compliance evidence.

Deployment shapes

Node classes. From **N0** (structurally minimal — no artifact host, no duties; microcontroller-class, TrustZone-M targets such as Cortex-M33/M55/M85) through **N1/N2** (phones, gateways, workstations) to **N3** (servers holding consensus roles), plus browser-class leaves that participate without carrying mesh duties. Absent features are *compiled out*, not switched off.

Topologies. A single vanilla broker with PolyCap clients; a micro-broker node stewarding local devices; a pure peer mesh; a **gateway** at a network boundary owning the actuation surface of a lab, plant, or fleet (the DMZ pinhole, made principled); a cloud backplane where servers run PolyCap for scale and multi-cloud neutrality.

Two ways to embed. An *embedded node* inside your application, or a *companion node* on the device shared by many applications — one node, one peer set, one policy, regardless of app count.

Hardware ladder. Linux first; Raspberry Pi-class hosts and RP2350-class leaves; Cortex-M55/M85 secure islands owning actuation peripherals; a licensable capability subsystem as the long-term target.

Integration surfaces

- **Any MQTT 5 broker** — no plugins, no modifications.
- **SDK in three levels:** a violation-free core, an ergonomic layer that compiles down to it with proven emission equivalence, and kits for common jobs (Share, Pay, Meet, Sync, Rescue, Gather).

- **Gateway product:** the reference monitor for device and instrument estates — MCP-style and CLI callers terminate at it; drivers receive sealed authority; receipts for every actuation.
- **Telemetry** flows into existing IoT pipelines unchanged.

Assurance posture

Eighteen ratified modules; a closed catalog of 437 typed errors; 177 tracked obligations; a byte-exact **conformance oracle** that grades implementations against deterministic vector corpora, regenerable from enumerated inputs on a bare machine; six designated public verification suites on the roadmap; a reference implementation graded against the oracle from its first commit — and, for the first module, **two independent implementations in two languages reproduce the reference corpus hash-for-hash.** Receipts are attributable, contemporaneous, original, and tamper-evident — the shape regulated-record regimes ask for (this is a design property, not a certification claim).

What PolyCap deliberately does not do

No global consensus ledger; no convertible token; no identity provider; no broker replacement; no machine learning inside the substrate; no unified device-health telemetry; no rule that requires trusting a node's own opinion of itself.

Status (September 2026)

Design set complete at v1.0. Reference implementation (**polyZero** runtime, **polyMesh** layer) under active development against the oracle; polyZero — the runtime for **N0**, the structurally minimal node class — implements the first three modules, all three independently reproduced; a test net is running, polyZero nodes talking to one another through a standard MQTT 5 broker with real-time OpenTelemetry tracing from the nodes themselves; walking skeleton and gateway demonstration scheduled for autumn 2026. Closed source; partners receive the conformance program today; public suite publication is on the roadmap.

Ideation prompts

1. Where does your infrastructure rely on **ambient authority** — an API key, a session, a broker ACL — that a compromised or confused component could wield in full?

2. Which **actuation paths** (valves, instruments, robots, payments) would you want a signed receipt for, every time?
3. Which devices or sites must **keep working when the network doesn't**?
4. Where do you currently run an integration project to move data between a lab and a plant, an edge and a cloud, a vendor and a customer — and would a shared authority model collapse it?
5. Which of your fleets is really a **multi-writer digital twin** waiting for per-branch write authority?
6. Where would **k-anonymous discovery** let you find capacity or expertise without exposing who has it?
7. Which services do you want **metered with evidence** rather than invoiced on trust?
8. Where would a **tighten-only policy overlay** let a business unit govern itself without weakening enterprise floors?
9. Which of your agent or driver artifacts would you deploy *today* if they could only ever hold the authority you handed them?
10. What would you build if authority were a thing you could **hand out, narrow, and take back** — with proof — the way you hand out keys?

Questions or partnership inquiries: Affectron, LLC.
