

eye3.ai VAOS Platform Overview

Vision-Native Agent Operating System

VAOS turns raw perception into structured intelligence and orchestrated actions. It is hardware-neutral, camera-multiplexed, and privacy-first.

Core Value Proposition

1. **Deterministic contracts** — Sensors emit the same JSON everywhere.
2. **Local-first autonomy** — Decisions happen at the edge, not the cloud.
3. **Composable agents** — Memory feeds reusable agent runtimes and action buses.

Architecture Flow

...

Sensors → Sensor Manager → Scene Contract → Memory → Agent Runtime → Action

...

Stage Responsibilities

- **Sensors:** Physical or simulated cameras delivering frames or detections.
- **Sensor Manager:** Normalizes input, attaches metadata (camera pose, zones, privacy flags).
- **Scene Contract:** Canonical JSON stored under `schemas/scene.schema.json`; enforces schema versioning and validation via AJV.
- **Memory:** Filesystem-backed store that tracks latest scene per camera plus rolling events/stats; future-ready for Postgres/MinIO adapters.
- **Agent Runtime:** Deterministic automations that subscribe to memory deltas and emit intents (alerts, PTZ moves, downstream workflows).
- **Action:** MQTT topics (`eye3/action/<agentId>`) or REST callbacks that effect change in the physical world.

Data Surfaces

- **MQTT:** Transport for scenes, events, captions, keyframes, and actions.
- **REST:** Queryable state for dashboards or downstream services.
- **WebSocket:** Push channel mirroring REST payloads for low-latency UI updates.

Deployment Stack

- **Broker:** Mosquitto (Docker) with topics defined in `docs/api/mqtt-topics.md`.
- **Ingest:** TypeScript service validating scenes/events via shared contracts.
- **API:** Fastify-based service exposing health/scenes/events/stats endpoints.
- **Dashboard:** Vite/React client that binds directly to the API semantics (no bespoke data shaping).

Use Cases

- Robotics fleets needing on-device perception and action loops.
- Industrial automation (safety zones, pick-and-place verification).
- Smart infrastructure (occupancy, queue length, anomaly alerts).

Stage 2 Hooks

- Keyframe topic + noop handler reserved so media pipelines can land without refactoring transports.
- Captions placeholder objects included in the schema with optional rendering in UI.
- README instructions kept in sync with the dev runner so new engineers can onboard in under 10 minutes.