# VAOS Edge Runtime Datasheet

## Snapshot

• **Product:** Vision Agent Operating System (VAOS)
• **Focus:** Local-first, privacy-preserving multi-camera intelligence
• **Release:** Stage 2 MVP

## Supported Platforms

• Rockchip RK3588 class SoCs
• ARM64 edge compute boxes with GPU/NPU acceleration
• Raspberry Pi 5 (developer / simulation mode)
• macOS + Linux hosts for hub services and simulations

## Interfaces

• **MQTT:** Primary transport for scenes, events, and keyframes
• **REST API:** `/health`, `/scenes/latest`, `/events/recent`, `/stats`
• **WebSocket:** Real-time push of scene deltas to dashboards

## Core Services

• **Sensor Manager:** Normalizes per-camera frames into canonical Scene JSON (runs on edge
• devices or simulators).
• **Ingest:** Validates Scene/Event payloads and persists to local storage; rejects
• malformed payloads with schema diagnostics.
• **Memory Store:** Filesystem-first ledger of scenes, events, stats with a future plug-in
• path for Postgres/MinIO.
• **Agent Runtime:** Hosts behavior trees / automations responding to memory via MQTT or
• REST subscriptions.
• **Action Bus:** Publishes automation outcomes back to MQTT (`eye3/action/<id>`) —
• placeholder in Stage 2.

## Deployment Modes

1. **All-in-one dev loop** — broker + publisher + ingest + API on a single laptop via `npm run dev:demo`.
2. **Split edge/hub** — sensor manager on RK3588, hub/ingest/API on a local NUC.
3. **Air-gapped** — MQTT broker and hub run offline; UI served locally.

## Security & Privacy

• No cloud dependency; all computation runs on devices under your control.
• Scene payloads can be privacy-stripped (blur regions, redact identities) before leaving
• the sensor node.

• TLS-ready MQTT configuration via `infra/mqtt/certs` (coming in Stage 3).

## Performance Targets (Stage 2)

• 5 cameras @ 10 FPS synthetic scenes on a Mac mini M2.
• Sub-500 ms ingest latency from publish → REST availability.
• Dashboard refresh via WebSocket < 150 ms after new scene persisted.

## Roadmap Hooks

• Keyframe payload reserved (`eye3/keyframe/<cameraId>`) with binary/object-store pointer
• support.
• Caption service placeholder tied to scene IDs.
• Action bus topic contract documented but unimplemented to keep Stage 2 focused.