

Counter Terrorism · Control Center — V1 Design

Eagle Eye · Intelligence as a Service · design doc v1 · 2026-09-26 · status: draft, nothing built

1. Purpose

Relay real-time signals to a distributed team, with each signal read by a model against a standing corpus and routed by webhook to whoever can act on it.

2. Scope

In scope (V1) — one ingest source end to end · retrieval-backed classification · rule-based routing · webhook dispatch with retries · a read-only console.

Out of scope (V1) — autonomous action of any kind, classified or non-public sources, multi-tenant access control, mobile clients, analyst workflow tooling.

Explicitly not a goal. The system ranks and routes; it does not decide. A human acts on every notification.

3. Components

Component	Responsibility
Ingest worker	Pull or receive feeds, normalise to one event schema, assign an idempotency key
Vector store	ChromaDB, one collection per source class; embeddings written on arrival, not batched
Enricher	LLM reads the event plus top-k retrieved context, returns structured JSON only
Route engine	Maps the verdict to targets by use case; rules are data, not code
Dispatcher	At-least-once webhook delivery, exponential backoff, dead-letter queue
Console	Live queue, event detail with the evidence that produced the verdict, audit trail

4. Data flow

Ingest → normalise → embed → retrieve top-k → classify → route decision → dispatch → audit log.

Every stage writes to the audit log. The log, not the console, is the source of truth.

5. Key decisions

- **The model returns JSON, never prose** — severity, entities, confidence, suggested route. Free text cannot be routed on.
- **Every notification carries its evidence.** The retrieved passages travel with the alert so a recipient can check the reasoning rather than trust it.
- **Confidence is a ranking, not a probability.** LLM self-reported confidence is uncalibrated; use it to order a queue, never as a threshold for action.
- **Idempotency keys on every event.** Feeds redeliver, and a replay that double-notifies is worse than a miss.
- **Human gate above a severity threshold.** High-severity verdicts queue for confirmation before dispatch; low-severity route automatically.
- **Routing rules live in config.** Adding a party or a use case must not require a deploy.

6. Risks

- **False positives are the dominant cost.** A channel that cries wolf gets muted, and the muting is what loses the real signal.
- **Third-party webhooks fail.** Assume non-delivery: retries, dead-letter, and a visible delivery state per alert.
- **Retrieval quality caps everything downstream.** A thin corpus produces confident, wrong context — measure recall before trusting verdicts.

7. Open questions

Which feeds, and at what volume · who the notified parties are and through what channels · acceptable end-to-end latency · retention period for signals and audit log · who owns the routing rules.

8. Milestones

M1 one feed ingested and stored · **M2** retrieval plus classification with evidence attached · **M3** routing and dispatch with retries · **M4** read-only console over the audit log.