SOLUTION ARCHITECTURE

Multi-Site Facial Recognition Attendance System

Edge-first attendance, presence and movement tracking across 15+ offices
Design rationale · decision records · diagrams · delivery plan

SCENARIO BY	Fahd Khan · Head of IT · Cymax Technologies
SOLUTION BY	Muhammad Bilal · Senior Full-Stack Engineer / Solution Architect
CONTACT	iamfullstackeng@gmail.com · +92 310 7100663
	linkedin.com/in/code-with-passion
DATE / VERSION	28 July 2026 · v3.0

Raw video never leaves the building. Attendance never stops. Recognition never depends on a third party.

1 Executive summary

This document answers the assessment scenario with a complete solution architecture, and it shows its working: every major choice appears as a decision record with the options considered, the numbers behind the decision and the trade-off accepted. A design is only as strong as the alternatives it survived.

The core decision: **recognition happens inside each office, not in the cloud.** Each site runs a small on-premise AI server that detects and recognises faces locally and transmits only lightweight encrypted events to a central cloud. Three consequences follow. Biometric data never leaves company premises. Every office keeps recording attendance through internet outages. And recognition never depends on a third party: inference runs on hardware the company owns, so no per-image external dependency exists.

Delivery is AI-accelerated: agentic coding tools operate inside the repository under senior review, compressing a conventional 24-week programme to 16 weeks. The compression applies only to work made of code; hardware installation, compliance and physical site work remain at full human speed, and the plan says so explicitly.

100,000x	0	Week 8	Week 16
BANDWIDTH REDUCTION AT THE EDGE	INBOUND PORTS OPEN AT ANY OFFICE	PILOT OFFICE LIVE	ALL 15+ SITES LIVE

2 Requirements and capacity estimation

Stated requirements. Multiple IP cameras per office across 15+ locations; automatic detection and recognition of employee faces; attendance capture on entry; tracking of movement between areas; measurement of total time in office; centralised management and reporting.

Implied requirements a production system must also satisfy. Offline resilience, privacy and biometric compliance, low bandwidth footprint, fast recognition at entrances, anti-spoofing, scalability to new sites, and full auditability of access to biometric data. Each is addressed explicitly and carries a measurable target below.

Non-functional requirement	Target	How the design meets it
Attendance availability	99.9%+, independent of internet	Edge autonomy: recognition and buffering continue through any outage (section 5.4)
Recognition latency	< 700 ms end-to-end at entrance	All inference local, TensorRT-accelerated, in-memory matching (section 4)
False accept rate (FAR)	< 0.1% after pilot tuning	ArcFace embeddings, threshold tuning on real office lighting, liveness gate
False reject rate (FRR)	< 2% after pilot tuning	5-10 enrollment images per employee, IR-capable entrance cameras
Event delivery	No loss, no double count	MQTT QoS 1 + durable edge queue + idempotent consumer on event UUID
Dashboard freshness	< 2 s from detection	WebSocket push from the ingestion path via Redis
Site recovery (RTO)	< 1 hour	Cold-spare edge unit per region, restored from cloud configuration
Audit	100% of biometric access logged	Immutable audit log written on every human read of biometric data

Capacity estimation. Working assumptions: ~3,000 employees, ~10 cameras per site, ~150 cameras total. Event volume: ~50,000 events/day company-wide, peaking near 20 events/second in the 08:00-10:00 entry window, trivial

for one NestJS consumer with 100x headroom. Data growth: ~200 bytes per event, ~10 MB/day, ~3.6 GB/year, so a single PostgreSQL instance holds a decade of history. Embedding store: 5,000 employees x 512 floats is ~10 MB in total and fits in memory on every edge device. Video, if streamed instead: ~600 Mbps continuous upload, the number that rules out cloud processing. When a system is sized honestly, most fashionable infrastructure turns out to be unnecessary; the decisions in section 6 follow from these numbers.

3 High-level design

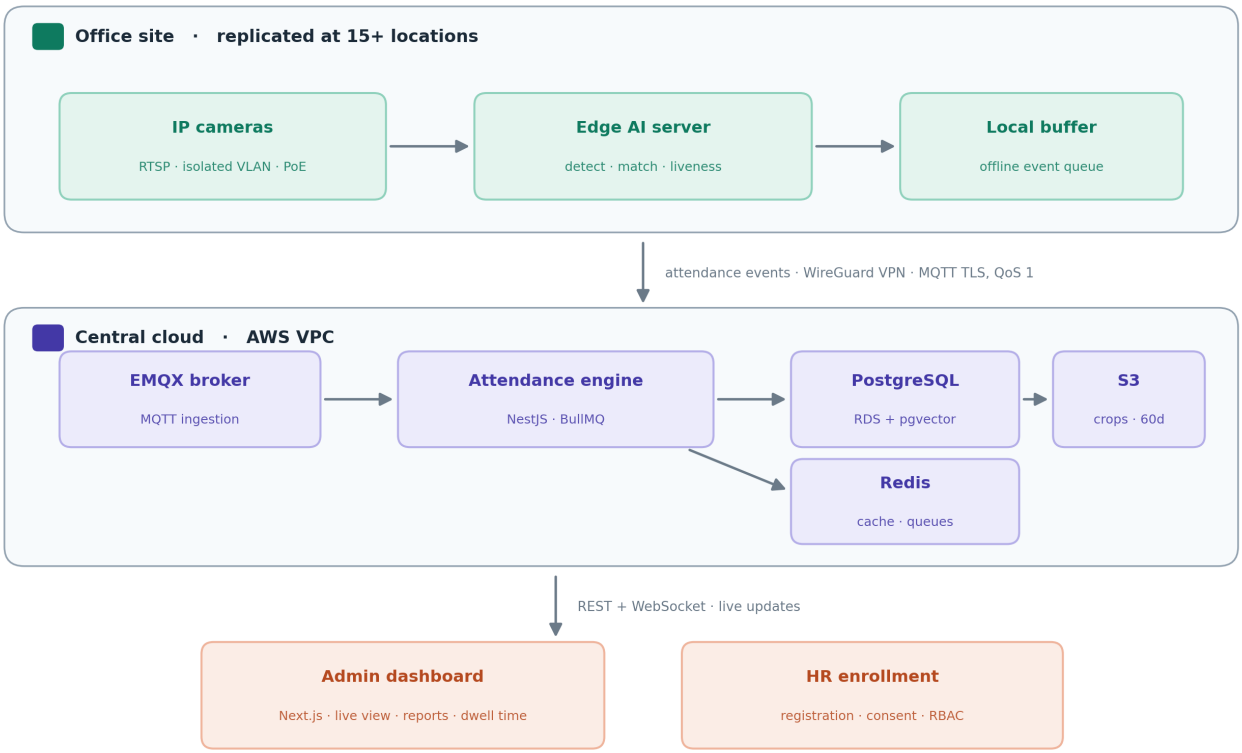

Figure 1. Three tiers: edge processing inside each office (green), central cloud consolidation (purple), applications (orange).

The flow in one paragraph. A camera sees a face. The edge server detects it, checks liveness, converts it to a 512-dimension embedding and matches it against the enrolled set in memory, entirely inside the office, in well under a second. The match becomes a small JSON event that travels through an encrypted tunnel to the cloud, where the attendance engine turns event streams into business facts: check-in, check-out, movement, dwell time. Managers watch it live on the dashboard. Everything else in this document exists to make that flow fast, private, resilient and cheap to run.

4 Recognition pipeline

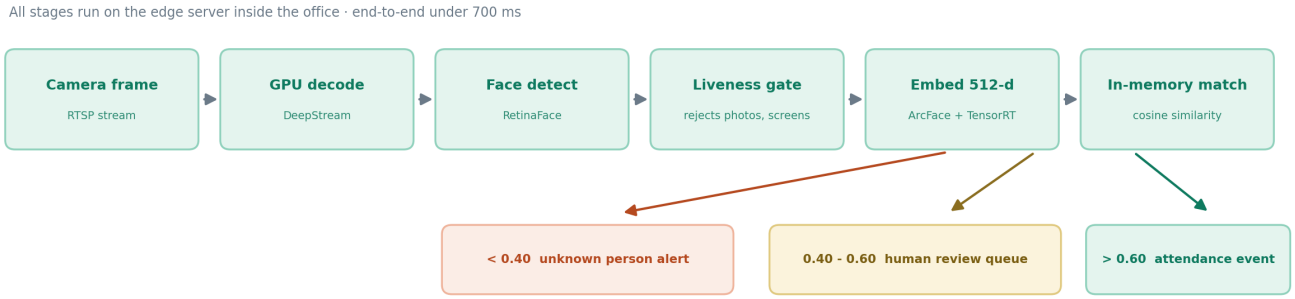

Figure 2. Per-frame pipeline on the edge server, with the three-way confidence branch.

A DeepStream pipeline decodes all camera streams on the GPU in a single process. RetinaFace localises faces per frame; a liveness model examines texture and motion cues to reject printed photos and phone screens before any matching occurs; ArcFace converts each live face into a 512-dimension embedding, accelerated by TensorRT. Matching is a cosine-similarity search against the in-memory enrolled set: above 0.60 becomes an attendance event, 0.40 to 0.60 is routed to a human review queue with the face crop attached, and below 0.40 raises an unknown-person entry for security. Both thresholds are tunable per site and are calibrated during the pilot against measured FAR and FRR, not guessed. A 30-second per-camera debounce suppresses duplicate detections of the same person, so one walk past a camera produces one event, not thirty.

5 System design

5.1 Edge layer, per office

PoE IP cameras stream RTSP on an isolated camera VLAN with no internet route; only the dual-homed edge server can reach them. The edge software runs as versioned containers (docker-compose) with signed over-the-air updates. Events publish over MQTT with QoS 1 through a WireGuard tunnel. A durable SQLite-backed queue holds every event until the broker acknowledges it. The device self-reports health metrics (stream status, GPU load, queue depth, temperature) so a degraded site is visible in Grafana within one minute.

5.2 Cloud layer

EMQX receives events from all sites. The NestJS attendance engine consumes idempotently: every event carries a UUID, so at-least-once delivery never double-counts. Business rules turn events into facts: the first entry-camera detection of the day opens check-in, the last detection closes check-out, zone-to-zone transitions accumulate into movement paths and per-area dwell time, and a site-timezone day boundary with an early-morning cutoff handles overnight shifts correctly. PostgreSQL (RDS) is the single system of record; Redis backs the live cache and BullMQ queues so reporting never competes with ingestion; S3 keeps only flagged crops with 60-day automatic deletion.

5.3 Data model

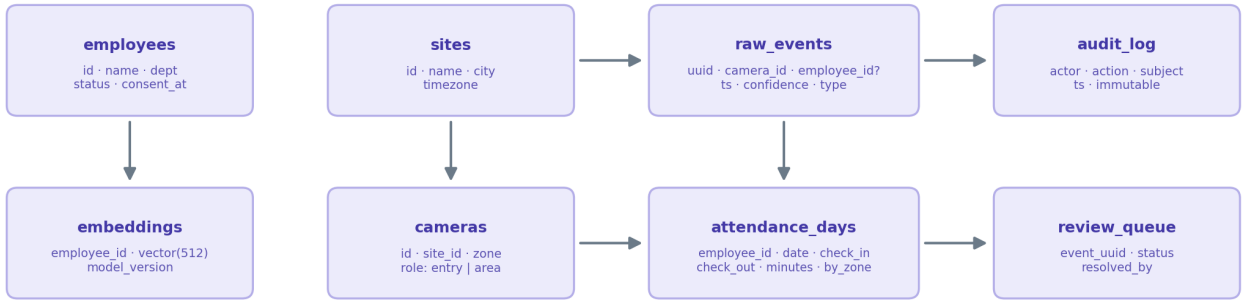

Figure 3. Core relational model. raw_events is append-only; attendance_days is derived and rebuildable.

The separation between raw_events and attendance_days is deliberate: raw events are immutable truth, while attendance is a derived view. If a business rule changes, for example a new dwell-time definition, the engine replays raw events and rebuilds attendance without data loss. Embeddings are versioned by model, so a future recognition-model upgrade re-enrolls transparently.

5.4 Offline resilience

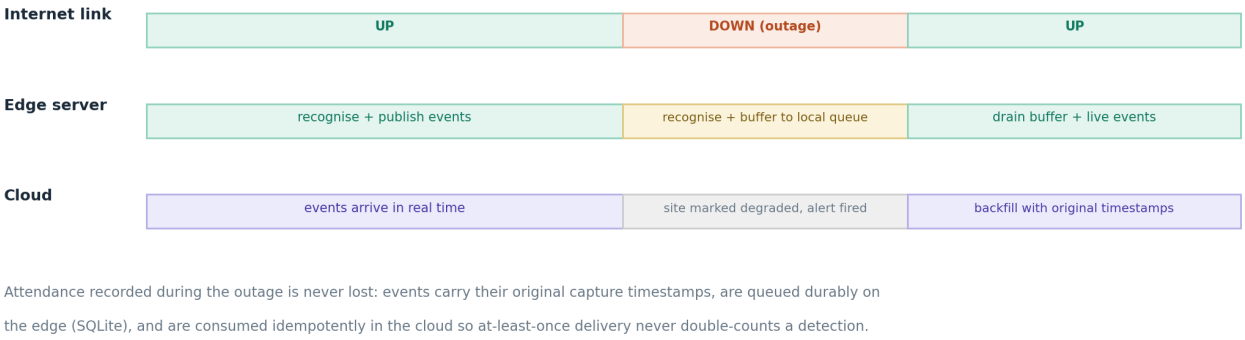

Figure 4. Behaviour across an internet outage: recognition never stops, events backfill with original timestamps.

5.5 API and interface design

REST resources: /employees, /sites, /cameras, /events, /attendance, /reports, /review-queue, versioned under /v1 with cursor pagination and RBAC-scoped responses. WebSocket channels per site and per dashboard view for live updates. MQTT topic scheme: site/{siteId}/events for detections and site/{siteId}/health for device telemetry, with per-site credentials so one compromised site cannot publish as another. All write paths accept idempotency keys.

5.6 Enrollment and lifecycle

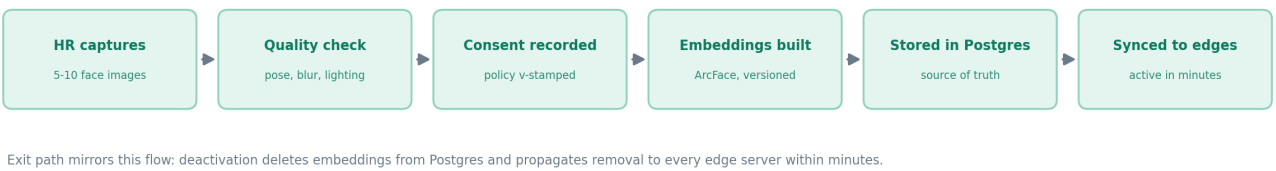

Figure 5. Enrollment flow; deletion on exit mirrors it and propagates to every edge within minutes.

6 Architecture decision records

Each record states the options considered, the decision, the reasoning with numbers, and the trade-off knowingly accepted.

ADR-1 · Where does video processing happen?

Option	Assessment
A · Stream all video to cloud	Simplest to build. But 150 cameras x 4 Mbps = ~600 Mbps continuous upload, cloud GPU cost at all hours, total attendance outage whenever a site loses internet, and raw biometric video leaving every office
B · Detect at edge, recognise in cloud	Cuts bandwidth but keeps the cloud on the recognition hot path: outages still stop attendance and face images still leave the premises
C · Full edge processing, events only	Recognition local; kilobytes per minute leave each site; offices fully functional offline; biometric data stays on premises. Requires one edge device per office

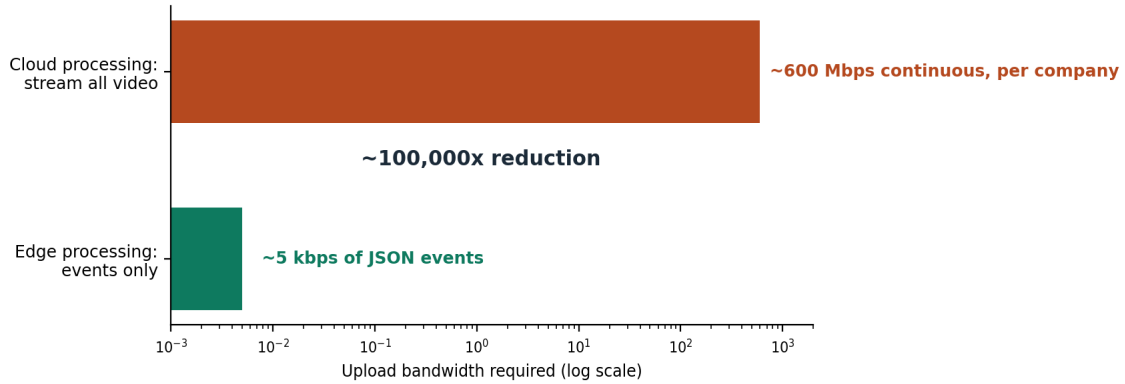

Figure 6. The deciding number for ADR-1, on a log scale.

> **Decision.** Option C. Bandwidth drops ~100,000x, availability decouples from internet uptime, and privacy exposure collapses. Trade-off accepted: we operate 15 edge devices; mitigated by identical container images, signed remote updates and cold spares.

ADR-2 · Recognition engine: build with open source, or buy a managed API?

Managed APIs bill per image. Across 150 cameras running continuously, that is a fee that recurs forever and grows with every camera added, while edge hardware is a one-time acquisition. It also sends every employee's face to a third party and makes recognition depend on the internet connection. The open-source alternative, InsightFace (RetinaFace for detection, ArcFace for embeddings), leads public accuracy benchmarks and runs entirely on the edge device. The chart below is the decision, drawn: shape only, no units.

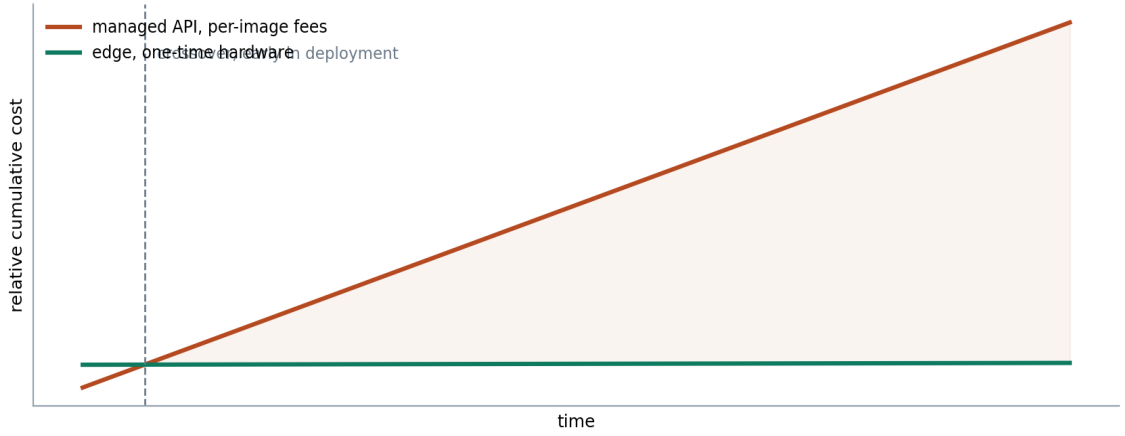

Figure 7. Relative cumulative cost over time, shape only. The managed line has no ceiling because it is billed per image; the edge line flattens once hardware is acquired.

> **Decision.** Build on InsightFace at the edge. No training is needed, only enrollment using pretrained models. Managed APIs suit low-volume bursty work; continuous video is the workload where per-image billing never stops. Trade-off accepted: we own accuracy tuning, with a dedicated pilot tuning window in weeks 9 to 11.

ADR-3 · Edge hardware

Option	Assessment
Raspberry Pi + USB accelerator	Cheapest but caps at 2-3 camera streams; a 10-camera office needs 4+ units, multiplying failure points and management burden
x86 workstation + RTX GPU	Strong performance, easiest to source in Pakistan and UAE, standard Linux tooling; higher power draw
NVIDIA Jetson Orin NX	Purpose-built for edge inference, 10-25 W, runs DeepStream multi-camera pipelines natively, comfortably handles 8-16 streams per unit

> **Decision.** Jetson Orin NX as primary, x86 + RTX as the approved procurement alternative so supply chain never blocks rollout. One device per office beside the PoE switch; both options run the identical containerised stack.

ADR-4 · Site-to-cloud transport

Option	Assessment
HTTPS polling / REST push	Simple, but delivery guarantees, retry and dedup must be hand-built; wasteful keep-alive traffic from 15 sites
Kafka clients at the edge	Kafka excels at datacenter-scale stream processing; heavy cluster clients at 15 small sites for a few hundred messages per minute is the freight train delivering envelopes
MQTT over TLS (EMQX broker)	Purpose-built for small messages on unreliable links: QoS 1 at-least-once delivery, persistent sessions that resume after outages, featherweight edge client

> **Decision.** MQTT for the last mile. Should heavy analytics pipelines arrive later, Kafka slots in cloud-side behind the broker without touching any office. Trade-off accepted: one additional component (EMQX), a single small container with a well-understood operational profile.

ADR-5 · Vector storage for face embeddings

The fashionable answer is a dedicated vector database (Milvus, Pinecone, Weaviate). The sizing from section 2 says otherwise: 5,000 employees x 512 floats is ~10 MB of vectors in total. Dedicated vector databases are engineered for billions of vectors; deploying one here adds an entire distributed system to operate, secure and back up, to solve a scale problem that does not exist. pgvector inside PostgreSQL answers exact nearest-neighbour search over this set in under a millisecond, and the embeddings are backed up, replicated and access-controlled with everything else.

> **Decision.** PostgreSQL + pgvector, with a further refinement: hot-path matching never touches the database at all. Each edge server holds its embedding set in memory and matches in microseconds; Postgres is the source of truth that syncs enrollments down. Matching at the edge, truth in the cloud.

ADR-6 · Cloud platform

Option	Assessment
Self-managed VPS	Full control and lowest cost, but backups, patching, failover and the compliance story for biometric data become in-house responsibilities
AWS managed services	RDS point-in-time recovery, S3 encryption and lifecycle policies, IAM audit trails, and ISO 27001 / SOC 2 certifications that matter when the data is biometric

Decision. AWS for a biometric system of record: managed durability and a certified compliance posture would cost far more to replicate in-house. The self-managed route remains documented as an option, with the responsibility transfer stated openly.

ADR-7 · Site connectivity and exposure

Options: a public MQTTS endpoint with client certificates (workable, but a public attack surface plus certificate lifecycle across 15 sites), classic IPsec site-to-site (mature, heavy configuration, brittle on dynamic IPs), or WireGuard tunnels from each office to the cloud VPC.

Decision. WireGuard. Roughly 4,000 lines of auditable code, faster than IPsec, survives the dynamic-IP behaviour of local office connections gracefully, and produces the strongest sentence a security review can hear: zero inbound ports open at any office, no camera or edge device ever visible from the public internet.

7 Network topology and security

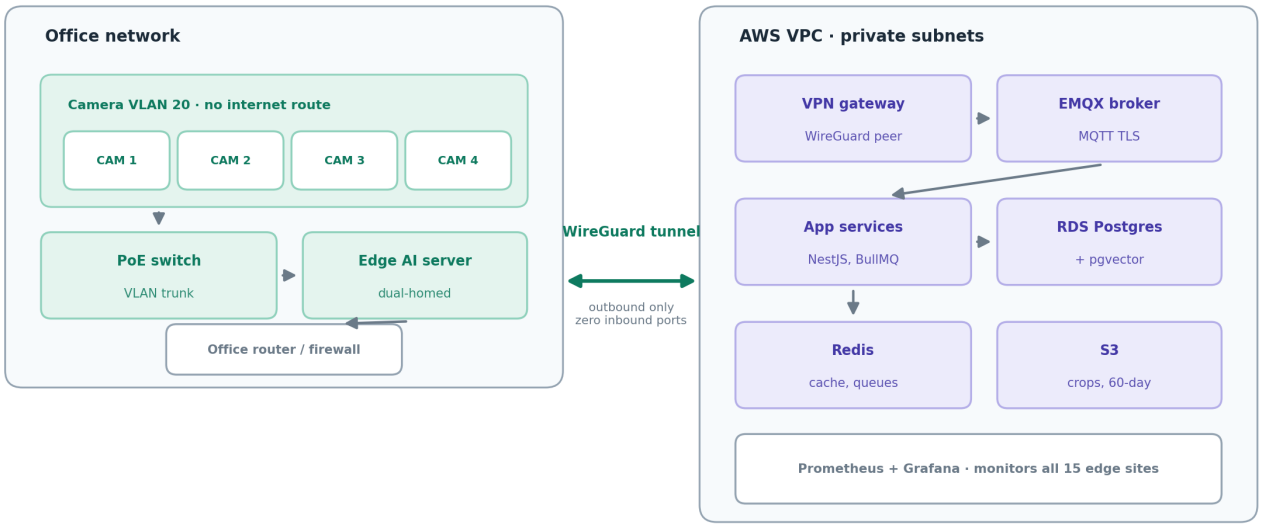

Figure 8. One office and the cloud VPC. Cameras are unreachable from the internet by construction, not by policy.

Layer	Measures
Network	Cameras on an isolated VLAN with no internet route, reachable only by the edge server. All site-to-cloud traffic inside WireGuard tunnels; zero inbound ports at any office. IP cameras are among the most compromised devices on the internet; VLAN isolation removes that risk class entirely
Data	TLS 1.3 in transit, AES-256 at rest (RDS, S3). Embeddings stored instead of images wherever possible; an embedding cannot be reversed into a face photo, materially reducing breach impact. Flagged crops auto-deleted after 60 days
Application	Short-lived JWT with refresh rotation; RBAC roles (HR admin, site manager, security, auditor); per-site MQTT credentials; API rate limiting; OWASP Top 10 review each release; immutable audit log of every human access to biometric data
Compliance	Written biometric policy; employee consent captured and version-stamped at enrollment; UAE PDPL and GDPR-aligned handling; documented retention and deletion, with exit-deletion propagated to all edge devices
Operations	Secrets in AWS Secrets Manager; signed over-the-air updates to edge devices; Prometheus and Grafana with degraded-site alerts inside one minute; automated encrypted backups with tested restores

8 Deployment and operations

Software delivery. Monorepo with GitHub Actions CI: every change runs the full test suite, builds container images, and publishes to a registry. Cloud services deploy on merge; edge devices receive signed updates in a canary sequence, one site first, then batches, with automatic rollback on failed health checks. Infrastructure is Terraform end to end, so the whole cloud environment is reviewable and reproducible.

Site provisioning playbook. A new office is a repeatable procedure, not a project: image the edge device from a golden image, register it with a one-time enrollment token, mount cameras per the placement standard, patch them into the camera VLAN, and the device pulls its configuration, embeddings and tunnel credentials from the cloud. Target: a two-person crew commissions a site in one day. This is what makes weeks 12-16 a rollout instead of fourteen small projects.

Observability. Grafana dashboards per site (stream health, GPU load, queue depth, event rate) and company-wide (attendance capture rate, review-queue depth, unknown-person rate). Alerts: site degraded, queue backlog growing, camera stream down, broker disconnect, abnormal unknown-face spikes. Every alert has a written runbook.

Quality and acceptance. Unit and integration tests generated and reviewed alongside every service; an end-to-end rig replays recorded RTSP footage through a real edge device into a staging cloud. Load tests at 100x expected event rate. A chaos drill severs a site's internet mid-day and verifies zero attendance loss after resync. Pilot acceptance criteria, measured over two weeks against a manual register: 98%+ attendance capture accuracy, FAR under 0.1%, FRR under 2%, and a passed offline drill. The pilot exits only on numbers, not on impressions.

9 AI-accelerated delivery

The engineering phases use agentic coding tools (Claude Code) operating inside the repository under senior-engineer direction and review. The claim is specific, not decorative:

Where AI accelerates	How	Typical saving
Service scaffolding	NestJS modules, DTOs, MQTT consumers, dashboard components generated against the approved architecture, reviewed line by line	40-60% of boilerplate time
Test suites	Unit and integration tests generated with every service; resync and idempotency logic gets exhaustive generated edge-case coverage	50%+ of test-writing time
Infrastructure as code	Terraform for VPC, RDS, S3 and monitoring; compose files and update tooling for edge devices	30-50%
Documentation	Runbooks, API references and site-installation playbooks drafted from the codebase, reviewed by the team	60%+

What AI does not compress, stated honestly: camera procurement and shipping, physical installation at 15 sites, compliance and consent review, pilot accuracy tuning against real lighting, and stakeholder decisions. The timeline keeps these at full human speed. **Governance:** every AI-generated line passes senior review before merge; security-sensitive code is additionally reviewed against OWASP guidance; CI gates every change. AI raises throughput; accountability stays human. Net effect: a conventional ~24-week programme delivers in 16 weeks with a leaner team.

10 Team and resources

Role	Count	Responsibility
Solution architect / tech lead	1	Architecture ownership, AI-workflow direction and review, delivery leadership
ML / computer-vision engineer	1	Model tuning, DeepStream pipeline, liveness, FAR/FRR targets

Role	Count	Responsibility
Backend engineer	1-2	NestJS services, MQTT bridge, attendance engine, APIs
Frontend engineer	1	Dashboard, enrollment module, live views
DevOps engineer	1	VPN mesh, CI/CD, monitoring; part-time after initial setup
QA engineer	0.5	Test strategy and pilot acceptance measurement
Field technicians	per city	Contracted camera mounting, cabling, rack installation

Per-site equipment: ~10 PoE cameras (IR-capable at entrances), one PoE switch, one edge AI device (Jetson Orin NX or x86 + RTX), mounting and cabling. One cold-spare edge device per region. Total engineering effort approximately 20 person-months with the AI-accelerated workflow, against roughly 30 conventionally.

11 Timeline

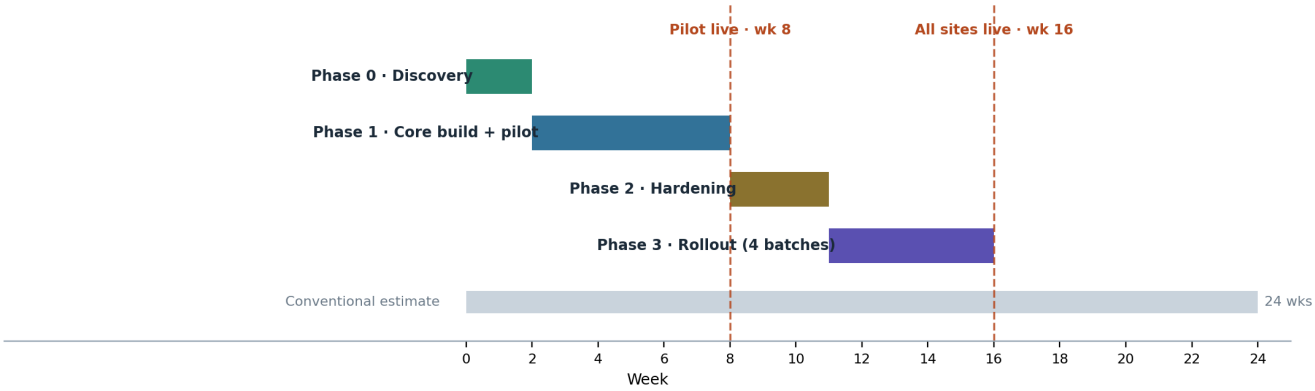

Figure 9. Sixteen-week programme with pilot and full-rollout milestones; conventional 24-week estimate shown for reference.

Phase	Weeks	Scope	Milestone
0 · Discovery	1-2	Parallelised site surveys, camera placement standard, biometric policy drafted and reviewed by counsel, architecture sign-off; hardware ordered day one	Approved design and rollout plan
1 · Core build + pilot	3-8	Edge pipeline, cloud services, dashboard MVP, enrollment; installation overlaps development; one office fully deployed	Pilot live with real attendance, week 8
2 · Hardening	9-11	Threshold tuning to FAR/FRR targets on pilot data, lighting edge cases, offline chaos drill, security review	Production readiness sign-off on measured numbers
3 · Rollout	12-16	Remaining 14 sites in four overlapping batches using the provisioning playbook	All 15+ sites live, handover and training

12 Assumptions and risks

Risk	Mitigation
Recognition accuracy in poor lighting	Camera placement standard, IR-capable entrance cameras, pilot threshold tuning with measured FAR/FRR acceptance gates
Spoofing (photos, screens)	Liveness detection before matching; flagged-event review queue for HR
Site connectivity loss	Durable local buffering with original timestamps; automatic resync; degraded-site alerting inside one minute
Regulatory change on biometrics	Embeddings-first storage, 60-day crop retention, tested deletion procedures
Edge hardware failure	Cold-spare per region; golden-image restore from cloud configuration in under one hour
AI-generated code quality	Mandatory senior review before merge; generated test suites; double review on security paths; CI gates every change

Assumptions: company headcount in the low thousands; each office hosts one small edge device in its network rack; entrance areas allow camera placement with adequate lighting.

13 Closing

Three commitments distinguish this design. **It shows its reasoning**: seven decision records carrying the alternatives that lost and the numbers that decided. **It is sized against reality**: capacity estimation first, technology second, which is why the stack is lean instead of fashionable. **It is engineered for the physical world**: offline resilience, a provisioning playbook, liveness against spoofing, and acceptance gates measured in FAR and FRR rather than impressions. I would welcome the opportunity to walk through any decision record in depth.

Muhammad Bilal · Senior Full-Stack Engineer / Solution Architect · iamfullstackeng@gmail.com · +92 310 7100663