Guard — Nigeria’s Citizen-First Emergency Alert Platform

A Technical Marketing Paper

Presented by: Office of the Chief Marketing Executive, Contrivances Engineering Ent. Ltd. Product: Guard — Emergency Alert System Audience: Government agencies, private security firms, fleet operators, community associations, developers, and citizens

---

Executive Summary

Every year, Nigerians lose lives, vehicles, and property because **by the time help is called, help is already too late.** Security threats, fires, medical emergencies, road accidents, and increasingly frequent banditry attacks all share the same bottleneck: a few critical minutes between “something is wrong” and “someone who can help knows about it.”

Guard is the system that closes that gap.

Guard is a cloud-native, mobile-first emergency alert platform that turns every smartphone into a two-way responder node. When a citizen raises an alarm, everyone within the relevant geographic radius — neighbors, police, fire service, government dispatchers — is notified in real-time via push notification, in-app alert, and WebSocket broadcast. Administrators see a live map, AI-generated intelligence summaries, escalation workflows, and complete audit trails.

Guard also bundles **ePass**, a premises vehicle access-control system that doubles as a passive stolen-vehicle detection grid: any registered vehicle flagged stolen is automatically alerted to its owner (with GPS and Google Maps link) the moment it is scanned at any premises or mobile checkpoint anywhere in the network.

And because a citizen-driven alert network is only as credible as its signal-to-noise ratio, Guard ships with a **peer-verification and anti-abuse system**: other users near an alert can confirm it as genuine or flag it as false; three false flags (or a single admin flag) instantly sanction the sender and block them from creating new alerts until a ₦10,000 fine is paid. Admins can override any sanction.

Guard is free for citizens. Our revenue model is a donation funnel powered by Flutterwave — the app never paywalls safety.

Guard sign-in page *Figure 1 — The Guard sign-in experience. Passwordless magic-link authentication. No passwords to forget, no accounts to hack. A “Support Guard — Donate” link on the login page lets supporters contribute without signing in.*

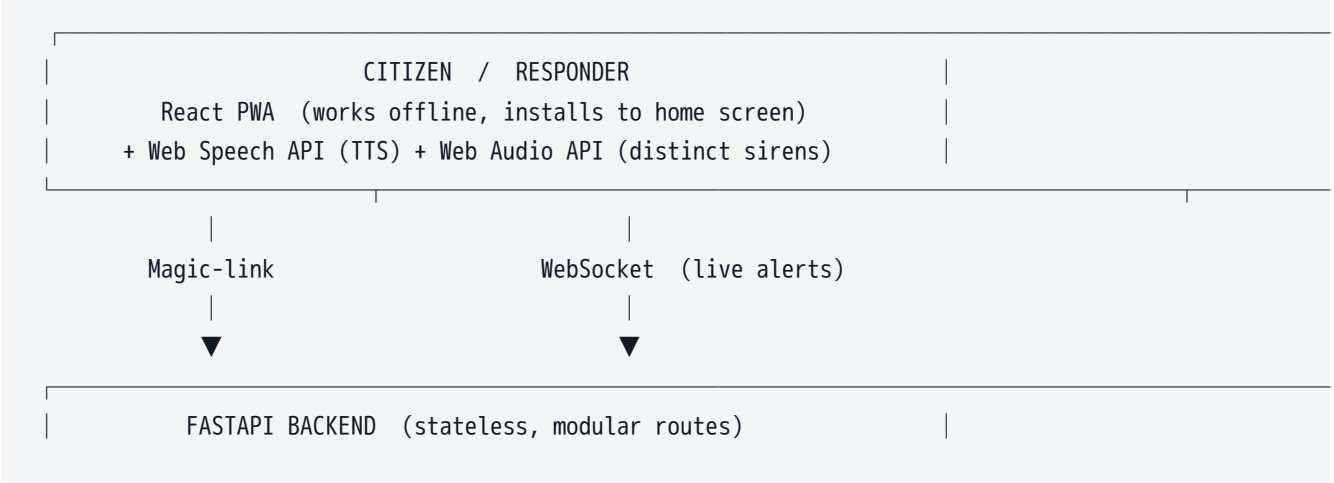
---

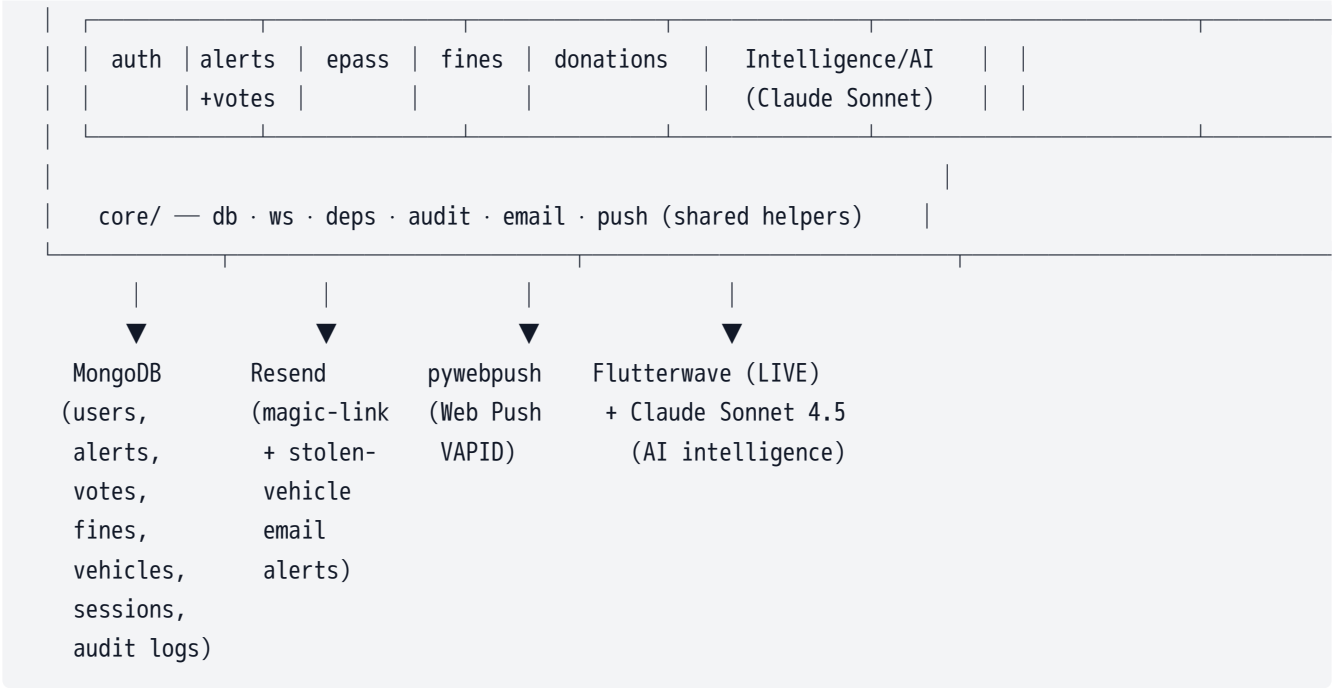
1. The Problem

Observation	Consequence
A typical emergency-call chain has **5–7 hops** (victim → friend → friend's friend → police operator → duty officer → patrol team).	Response times of 20–90 minutes in urban centers; often hours in rural areas.
Response teams often **don't know the exact GPS of the incident** — only the general area.	Teams drive to the wrong block, waste minutes.
Neighbors and bystanders, who could help in under 60 seconds, never learn the incident is happening.	Lives and property lost that proximity-alerted citizens would have saved.
Vehicle theft recovery rates in Nigeria are under 5%. Most stolen cars are re-plated and resold within 48 hours.	Billions of Naira annually lost. No passive detection mechanism exists.
Government intelligence units rely on manual incident aggregation. Patterns emerge days or weeks late.	Hotspots, repeat offenders, and attack patterns go un-mapped.
Banditry attacks in North-West Nigeria routinely involve 50+ attackers. A 10-km alert radius is too small — residents 15 km away in a neighboring village will be targeted next.	Attacks cascade unchecked.
Road hazards (potholes, collapsed bridges, men-at-work, accidents, flooded roads) cause secondary accidents every day. Drivers have no way to pre-warn oncoming traffic.	Repeat crashes at the same blackspots.
Crowdsourced alert networks everywhere get gamed by pranksters, competitors, and bad actors. Without peer-verification, false alarms erode public trust and drain responder attention.	Real emergencies get ignored alongside the noise.

Guard is architected as a direct response to each of these seven failure modes.

2. How Guard Works — High-Level Architecture





Key design choices:

- **PWA-first.** Guard runs in any modern browser. No app-store gatekeeping. Installable to Android/iOS home screens. Works offline with service worker fallback.
- **Passwordless auth.** Magic-link email sign-in via Resend. No passwords to steal, no "forgot password" flow. Admin role auto-assigned from email domain (`.contritrack.com` → system admin, `.gov.ng` → government admin, e.g. `@parastatal.gov.ng` is recognized).
- **Real-time fan-out.** New alerts are broadcast over WebSockets to every connected client and via Web Push (VAPID) to subscribed devices — even when the app is closed.
- **Role-based access.** Three roles: Citizen, System Admin, Government Admin. Each sees different views and can take different actions.
- **Free for users. Donations fund it.** No paywall, no ads, no subscription. One Flutterwave button in the top-right nav.

---

3. Feature Deep-Dive

3.1 Seven Alert Types — Covering Every Nigerian Emergency

Guard dashboard with seven alert triggers, live map, and proximity sliders *Figure 2 — The citizen dashboard. Seven emergency buttons in an equal-width horizontal scroll strip (left on desktop, across the top on mobile), a live map with proximity radius overlays (center), and a real-time alert feed with inline peer-verification buttons (right). Every alert is one tap away.*

Type	Color	Icon	Default Radius	Behavior
Security	Red	Shield-alert	10 km	Fan-out + escalation + push
Fire	Orange	Flame	10 km	Fan-out + escalation + push

Type	Color	Icon	Default Radius	Behavior
Medical	Blue	Heart-pulse	10 km	Fan-out + escalation + push
Accident	Yellow	Car	10 km	Fan-out + escalation + push
Banditry	Dark rose	Warning-triangle	**20 km** (larger because banditry attacks cascade)	Fan-out + escalation + push, **police-siren audio cue**
eCaution (road hazard)	Amber	Traffic-cone	**500 m** (narrow — affects only oncoming traffic)	Fan-out + push, **description read aloud via Text-to-Speech** so drivers hear it hands-free
I'm Here (Location share)	Emerald	Map-pin	n/a	Silent broadcast — shown on map, **no notifications to others**, exempt from peer-voting

Why this matters: - **Banditry** gets a 20-km radius because in North-West Nigeria, attackers frequently move between villages. Alerting the next target before attackers arrive is a direct life-saver. A distinct **police-siren audio cue** ensures recipients cannot miss it. - **eCaution** is Guard's road-safety layer. Drivers who spot "men at work," "collapsed bridge ahead," "pothole," "accident on road," "flooded road," "fallen tree," "livestock on road," "railway crossing," "bumps ahead," or "police checkpoint" can fire a one-tap alert. Every other driver within 500 m hears a short attention chime followed by the caution spoken aloud via the browser's **Web Speech API** — eyes stay on the road, hands stay on the wheel. This has the potential to prevent the daily cascade of secondary crashes at Nigerian blackspots. - **"I'm Here"** is a non-emergency share — use it to say "I've arrived safely" to family, or "I'm at the scene" when responding to help. It shows on the map but does NOT spam every user's phone. - **Per-user proximity sliders** let each citizen choose their own radius per alert type (1–100 km for emergencies, 100 m – 2 km for eCaution). Rural users who want to help neighbors can set 50 km; urban users may prefer 5 km.

Technical note. Alert creation at `POST /api/alerts` is intentionally **the only endpoint with a hard safety guarantee — no paywall.** The only restriction is anti-abuse (see §3.4 Peer Verification).

3.2 Live Map with Proximity Heat Rings

Every active alert renders on an interactive Leaflet map with a colored circle showing its proximity radius. Admins can toggle a **heatmap view** to see density patterns over time — essential for government intelligence units identifying hotspots. The map ships a **dark-mode tile set** (CartoDB dark) that matches the app's dark theme and is easier on night-shift responders' eyes.

- Tap any alert in the feed → map flies to its location
- Dashed proximity rings show the user's own configured radius per alert type (great for "will I be notified if an alert happens *there*?")
- Real-time updates via WebSocket — new alerts appear within 1 second of reporting

---

3.3 ePass — Premises Access Control + Passive Stolen-Vehicle Grid

ePass vehicle registry *Figure 3 — ePass vehicle registry. Owners mark vehicles as "Good" or "Stolen." Every scan at any premises is logged, searchable, and contributes to the national stolen-vehicle detection grid.*

ePass has six complementary modules, accessible as tabs:

1. **Vehicles** — Register/transfer/manage vehicles. Each gets a unique QR code.
2. **Scan** — Point your phone at a QR code at a premises gate. Instant read-out: GOOD / DENIED / STOLEN. Stolen plates auto-trigger a security alert.
3. **Premises** — Register fixed premises (office gates) OR **mobile checkpoints** (police roadblocks — GPS is auto-captured from the scanner's device).
4. **Custom Access Criteria** per premises — allow-list, block-list, and time-window (e.g. "only these 10 plates, only 06:00–22:00"). Deny reasons are shown in scan results and history.
5. **History** — Complete scan audit trail with GPS, timestamp, result, denial reason.
6. **Fleet Management** — Organizations bulk-manage fleets of 10–10,000 vehicles. CSV import for onboarding. One-click bulk status change (mark all "stolen" if a depot is compromised).

The killer feature: passive stolen-vehicle detection.

Imagine a thief steals a car from Ibadan. Owner marks it stolen in the Guard app. Two days later, an office in Abuja scans that same plate at their gate — **the scanner immediately shows "STOLEN"**, a security alert is fired with GPS coordinates, AND the owner receives an automated email:

> *"Great news — we found your vehicle! …We found your vehicle. Buy us wine. Let's celebrate! 🍷"*

The email includes a Google Maps link to the exact location. No coordination with law enforcement is required — the owner can act immediately. **This turns every Guard user into a passive recovery sensor.**

3.4 Peer Verification & Anti-Abuse Sanctions

A citizen-driven alert network only works if the signal is trustworthy. Without peer-verification, a handful of pranksters, competitors, or bad actors can flood the feed with fake fires and bandit attacks — causing responders to tune out and real emergencies to be missed.

Guard's solution is a three-tier trust layer baked into the alert lifecycle.

Tier 1 — Proximity-gated voting. Every alert card has inline **True / False** buttons. To vote, a user must: - Be **within 500 m** of the alert's GPS coordinates (backend-enforced via Haversine distance — spoofed GPS is still spoofed, but you have to physically be there) - Be **a different user** from the sender (you cannot vote on your own alert) - **Vote once** per alert (one decision, no flip-flopping)

Alerts display live counts: `👍 n` `👎 n`. The `I'm Here` location-share alerts are **exempt** from voting — they're presence broadcasts, not emergency claims.

Tier 2 — Automatic sanction threshold. Sanctions trigger when either of two conditions is met: - **3 citizen users** cast `False` votes on the same alert, OR - **1 admin** (system or government) casts a single `False` vote (admin override + authority)

When triggered: - The alert gets a black **FLAGGED FALSE** badge across every user's feed in real-time (WebSocket broadcast) - The **sender is sanctioned** — their `sanctioned` flag is set in the database - The sanction is recorded in the audit log with both the flagging users and the admin (if any)

Tier 3 — Fine-and-restore. A sanctioned user can still: - Sign in normally - View the map and alert feed - Resolve their own existing alerts - Browse all other features

A sanctioned user **cannot**: - Create new alerts (`POST /api/alerts` returns `403` with the fine message)

To lift the sanction, the user pays a **flat ₦10,000 fine** via Flutterwave. A red banner at the top of their dashboard shows the **Pay ₦10,000** button; clicking it redirects to Flutterwave's hosted checkout. On successful payment: - The webhook (`/api/fines/webhook`) or client-side redirect-verify endpoint updates the fine to `paid` - The sanction is lifted (`sanctioned` flag cleared) - The user immediately regains alert-creation ability - The whole transaction is audit-logged and idempotent (replay-safe)

Admin override. System and government admins can clear any sanction without payment via `POST /api/users/{user_id}/clear-sanction` (useful for genuine emergencies mistakenly voted false; logged with the admin's ID).

Why this design: - **500 m radius** is tight enough that only people who could realistically witness the incident get a vote. A false security alert in Lagos cannot be downvoted by someone in Abuja. - **₦10,000 is calibrated to sting without being ruinous.** It's ~½ a day's minimum wage in urban Nigeria — high enough to deter serial abuse, low enough not to lock out a first-time mistake permanently. Once paid, the user walks away with a real lesson about crying wolf. - **The fine funds the platform.** Every fine collected flows into the same account pool as donations — abusers literally subsidize the safety net they tried to corrupt. - **Admin single-vote authority** matters because some false alerts are obvious (e.g., an admin physically at the scene of what was reported as a fire and seeing no fire). Requiring a 3-user quorum in those cases would slow response.

This is, to our knowledge, **the first production emergency-alert app in Africa with a self-enforcing credibility layer.** Without it, the network is a toy. With it, the signal stays clean.

3.5 AI-Powered Intelligence Reports

Guard Intelligence page *Figure 4 — AI-powered intelligence reports. One click generates a summary of incident patterns, hotspots, and risk levels using Anthropic's Claude Sonnet 4.5.*

For government admins and system admins only: - **Dashboard Summary Report** — one-click AI analysis of all alerts in the last 30 days (trends, hotspots, risk levels, recommendations) - **Per-alert deep-dive** — pick any incident; AI writes a narrative report with recommended actions - **History tab** — all generated reports saved and browsable

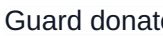
Powered by **Anthropic Claude Sonnet 4.5** via Emergent's managed LLM gateway — no OpenAI API key required from the operator.

3.6 Escalation Workflows + Admin Audit Trail

Alerts that remain unresolved automatically escalate through three levels based on per-type SLAs. Admins can: - Assign alerts to specific responders - View complete audit log of every action (alerts, votes, sanctions, fine payments, vehicle transfers, premise edits, sign-ins) - Filter by user, type, time, location - Review **Recent Sign-ins** (IP, device, browser, timestamp) to spot suspicious access patterns - Clear wrongly-triggered sanctions with a single API call (UI button coming next)


Recent sign-ins admin panel *Figure 5 — Recent sign-ins panel. Every login is captured with the user's IP (through X-Forwarded-For behind our Kubernetes ingress), parsed browser/OS, and timestamp. Critical for security auditing on an emergency-response platform.*

3.7 Donations & Fines via Flutterwave (LIVE)


Guard donate page *Figure 6 — The public donation page. Custom amount (₦100 minimum), "Donate anonymously" toggle, public counter, recent donors feed. Secure payment via Flutterwave — card, USSD, bank transfer.*

Guard now runs on **Flutterwave live credentials**, processing real Naira transactions across two distinct rails:

1. Donations (optional, any amount). - Public `/donate` page, anonymous-toggle supported - ₦100 minimum, hosted Flutterwave checkout (card / USSD / bank transfer) - Server-side verify via `/v3/transactions/:id/verify` on redirect + webhook safety net - Public counter + recent-donors feed for transparency

2. Sanction fines (₦10,000 fixed). - Triggered only when a user has been peer-verified as posting false alerts - Same Flutterwave gateway, separate `guard_fine_*` transaction references, stored in a dedicated `fines` collection for clear accounting - Idempotent — a replayed webhook cannot double-process or re-charge - On success, sanction is auto-lifted in the same transaction

Why donations, not subscriptions? - Emergency services historically are public goods. Paywalling them kills the network effect. - Every Naira donated is publicly accounted for (public counter on `/donate`). - Donations power infrastructure: servers, SMS credits, AI inference. - **Transparency = trust.** Every verified donation updates the "Raised so far" counter in real time.

Unified security posture. Both donation and fine flows: - Validate the HMAC `verif-hash` header on webhook calls (constant-time compare via `hmac.compare_digest`) - Re-verify every transaction server-side against Flutterwave's authoritative API (never trusting the client) - Require `currency = NGN` and `paid ≥ expected` before transitioning state - Log every successful payment to the audit trail - Are **idempotent** — replayed webhooks cannot double-count or double-lift a sanction

---

3.8 Scan2Call — Video Doorbell + Micro-Billing

Scan2Call turns any Guard user's front door into a QR-driven video intercom. Visitors scan a physical (or printable) QR sticker → the `/d/:qrToken` page opens on their phone → an Agora RTC audio+video call

auto-rings the resident anywhere in the Guard app. Unanswered rings roll to a 30-second **voicemail** delivered as an unread badge on the resident's floating Scan2Call widget.

Business model. Residents own a Naira-denominated Scan2Call wallet. Calls are billed **₦30/minute**, first minute debited on Answer, subsequent minutes debited every 60 s by a **persistent, restart-safe billing watchdog**. Wallets are recharged via Flutterwave. Unanswered rings and voicemails are **free**. When the wallet drops below the next minute the call auto-ends with a friendly "wallet_empty" reason so the resident is never surprised. A **15-second refund window** issues a first-minute refund whenever an answered call fails to connect (typical cause: cellular Agora join failure).

Onboarding hook. Every new signup receives a **₦100 welcome credit** — enough for one full minute — so first-time users experience the feature end-to-end without needing to top up.

Resident UX. A live *funds remaining* banner during the call shows funded talk-time and a per-second countdown to the next ₦30 debit; it flips to amber warning below one minute of runway. The visitor's derived location ("Calling from Akoka, Lagos" from browser GPS reverse-geocoding, or IP-based city as fallback) is shown before Answer. Optional deterrent sounds — real MP3 dog barks + a synth siren — can be played into the visitor's audio channel instead of engaging. An optional **"Show face"** toggle publishes the resident's camera to the visitor mid-call.

Reliability. The billing watchdog stamps `last_tick_at` on every debit and a startup sweep retro-bills any minutes elapsed during a FastAPI restart — closing an entire class of "container restart = free talk time" bugs. Unique `tx_ref = call_<ring_id>_min<N>` guarantees idempotency.

Abuse controls. The public visitor endpoint is rate-limited to **5 rings per 10-minute window per (visitor IP, QR-token)** so a leaked sticker photo can't be weaponised into a wallet-drain DoS. Voicemails per resident are capped at **20 per 24-hour period** to bound Mongo footprint.

Viral loop. The visitor's "Call ended" screen shows a UTM-tagged **"Get your own Guard"** link (`?utm_source=scan2call&utm_medium=visitor_end&utm_campaign=organic_referral`) so signups sourced from the QR-sticker viral loop are fully attributable in the marketing analytics stack.

Physical stickers. Residents can order a weatherproof QR sticker shipped to their premises via Flutterwave checkout. `@contritrack.com` system administrators bypass payment for internal fulfilment.

---

4. Security Posture

Layer	Mechanism
Authentication	Magic-link, single-use, sha256-hashed tokens in DB, 15-min TTL with auto-cleanup (MongoDB TTL index), rate-limited 3 requests per email / 10 min
Session	httpOnly, secure, samesite=lax cookies, 7-day expiry, server-side revocation on logout
WebSocket	Handshake requires a valid session cookie (or `?token=`), closes with code 4401 on unauth; per-user routing via `send_to_user(user_id, msg)` so private events (Scan2Call Agora publisher tokens, per-user voicemail notifications) never leak to another socket; `broadcast()` reaches only authenticated clients

Layer	Mechanism
Role assignment	Email-domain based, case-insensitive, subdomain-aware (`@dept.gov.ng` → govt admin)
Alert integrity	Peer-vote proximity gate (500 m, server-side Haversine), one-vote-per-user, self-exemption, location-share exemption — prevents remote gaming of the trust layer
Sanctions	Hard `403` on `POST /alerts` until fine is paid; admin override audit-logged; fine-payment idempotency matches donation flow
ePass scans	Stolen-vehicle detection wins over custom access rules (safety priority)
Scan2Call rate limits	5 rings / 10 min per (visitor IP, QR-token); 20 voicemails / 24 h per resident; wallet auto-end when credit runs out
Audit	Every auth event, alert, vote, sanction, fine payment, vehicle status change, premise edit logged with actor, target, IP, User-Agent, timestamp
Payments (donations + fines + wallet top-ups)	Flutterwave webhook HMAC signature verification via constant-time `hmac.compare_digest` ; server-side transaction re-verification; currency + amount guards; idempotent state transitions
CORS	Explicit production allow-list (`contritrack.com` , `www.contritrack.com` , Emergent deploy URL, preview URL) — no wildcard
Email	Resend with verified DNS + SPF/DKIM on `contritrack.com`
DB indexes	`created_at` descending on alerts, vehicle_scans; TTL on magic_tokens (auto-cleanup); votes deduplicated on `(alert_id, voter_id)` at query time

---

5. Deployment & Scale

- **Stack**: React 19 + Tailwind + Shadcn/UI (frontend); FastAPI + Motor (async MongoDB driver) + Pydantic (backend); Leaflet + leaflet.heat (maps); html5-qrcode (QR scanning); pywebpush (VAPID push); Resend (email); Flutterwave LIVE (donations + fines); Anthropic Claude Sonnet 4.5 (AI)
- **Frontend platform APIs**: Web Speech API (eCaution TTS), Web Audio API (distinct alert sirens including police-siren for banditry), Web Push API (VAPID subscription)
- **Backend layout**: modular — `server.py` is a thin 74-line orchestrator; business logic split into `core/*` shared helpers (db, websocket manager, auth deps, audit, email, push) and `routes/*` per-domain routers (auth, alerts, ePass, donations, fines) for clarity, testability, and parallel development
- **Hosting**: Kubernetes pod on Emergent's platform with preview & production URLs
- **Database**: MongoDB with proper indexes — tested with 45+ active alerts rendering instantly; 61/61 backend unit tests green
- **Scale ceiling (current architecture)**: comfortably 100,000 active users; 1M alerts/day; geographic fan-out is O(N log N) via proximity pre-filters

---

6. Benefits by User Segment

For **citizens**

- Emergency reporting in **two taps** (type + confirm)
- Real-time alerts for incidents within your chosen radius — protect your community
- Passive stolen-vehicle recovery for free
- **Peer-verification layer** means the alerts you receive are credible — no crying wolf
- Zero cost, no ads, no subscription pressure, passwordless sign-in

For **drivers and road users**

- **eCaution** broadcasts road hazards (men at work, collapsed bridge, pothole, flooded road, livestock, fallen tree, accident ahead, bumps, rail crossing, police checkpoint) to every vehicle within 500 m
- Warnings are **read aloud via Text-to-Speech** — eyes stay on the road, hands stay on the wheel
- Default 500 m radius is engineered so only oncoming/nearby traffic is notified (no spam)

For **private security firms / corporations**

- ePass turns every corporate gate into a stolen-vehicle sensor
- Fleet management onboards hundreds of vehicles via CSV
- Custom access criteria per premises (allow-list, block-list, time windows)
- Full audit trail for compliance
- Custom branding available on enterprise tier (future roadmap)

For **government agencies / police / parastatals**

- Real-time heatmaps of incidents
- AI intelligence reports show patterns your manual analysts miss
- Assign/escalate workflow for dispatch teams
- **Single-admin-vote sanction** gives verified officers authority to immediately suppress false alerts and sanction serial abusers — critical during riots, elections, and attack cascades when misinformation spikes
- Every officer login captured with IP + device
- Citizen-first network-effect: more users = richer signal

For **fleet operators**

- Instant stolen-vehicle detection across the Guard premises network
- Bulk onboarding (upload CSV of 500 vehicles, done in 30 seconds)
- Owner auto-notified with GPS + Google Maps link on detection

For community associations

- Neighborhood-scale proximity radius (1–5 km) keeps alerts hyper-local
- Mobile checkpoint feature turns volunteer patrols into Guard sensors
- “I’m Here” enables safe-check-in during incidents
- Peer-verification means the youth group cannot spam the WhatsApp-adjacent feed with pranks

---

7. What’s New in v7 — July 2, 2026

This release turns every Guard user’s front door into a smart, revenue-generating video doorbell — **Scan2Call** — and hardens the platform’s real-time layer against a class of anonymous-eavesdropping bugs uncovered during a mid-cycle security audit. Nine of the paid or engineering improvements below are direct responses to production feedback from the first weeks of live deployment on `contritrack.com`.

7.1 Scan2Call — Video Doorbell for Every Guard User

Any user can now order (or print) a **physical QR sticker** for their gate. Visitors scan the QR with any smartphone → the `d/:qrToken` page opens → they are auto-connected to the resident’s phone over an **Agora RTC audio/video call**. The resident answers from a floating Scan2Call widget available anywhere in the app; unanswered rings roll to a 30-second **voicemail** delivered as an unread badge on the widget.

Business model. Each resident owns a **Scan2Call wallet**. Calls are billed **₦30 per minute** — first minute debited on Answer, subsequent minutes debited by a persistent server-side billing watchdog every 60 s. There is **no upper time limit** — calls run as long as the wallet has credit; when it runs dry the call auto-ends with a friendly toast (“wallet ran out of credit — top up to keep talking”). Voicemails and unanswered rings are **free**.

Onboarding hook. Every new signup receives a **₦100 welcome credit** into their Scan2Call wallet — enough for one full minute of a live call, without asking for any payment. First-time users can experience the feature end-to-end before deciding to top up.

UX polish. Deterrent picker (real MP3 dog barks + synth siren) so a resident can bark at a suspicious visitor instead of engaging. Optional **“Show face”** toggle publishes the resident’s camera to the visitor (off by default). Live **“funds remaining” banner** during a call shows exactly how many funded minutes are left + a per-second countdown to the next debit; goes amber below one minute of runway. The visitor’s **city or precise street address** (“Calling from Akoka, Lagos”) is shown to the resident before they answer — computed from browser GPS when permitted, or from IP-based city lookup as a fallback.

Reliability. A persistent Mongo-backed watchdog survives FastAPI restarts and retro-bills missed minutes so a mid-call container restart cannot let the visitor talk for free. Failed Agora connects trigger an automatic **first-minute refund** inside a 15-second window. Public visitor endpoints are rate-limited to **5 rings per 10-minute window per (IP, QR-token)** to keep a leaked QR photo from being weaponised into a wallet-drain DoS.

Post-call marketing loop. The visitor's "Call ended" screen shows a UTM-tagged **"Get your own Guard"** link (`?utm_source=scan2call&utm_medium=visitor_end&utm_campaign=organic_referral`) so we can measure the QR-sticker viral loop in analytics from day one.

7.2 Physical Sticker Ordering via Flutterwave

Residents can order a weatherproof QR sticker shipped to their premises. Checkout goes through Flutterwave (delivery address + phone captured at order time). System-administrator accounts (`@contritrack.com`) bypass payment for internal fulfilment. Status is tracked on-app (pending → paid → shipped → delivered).

7.3 Security Hardening — WebSocket Authentication

The mid-cycle **security audit** flagged one HIGH finding: `/api/ws` was accepting anonymous connections and firehose-broadcasting every user's events — including Agora doorbell publisher tokens (allowing live-call eavesdropping) and live GPS coordinates from emergency tracks. This has been fully closed: - `/api/ws` now requires a valid `session_token` cookie at handshake and closes with code 4401 otherwise. - The connection manager was rewritten with per-user routing (`send_to_user(user_id, msg)`). All 5 Scan2Call WS emits (ring, end, voicemail, watchdog force-end, reconciler) now go **only** to the resident who owns the ring — the Agora publisher token never touches another user's socket again. - Broadcast events (new alerts, admin sanctions, tracking updates) still fan out network-wide but only to **authenticated** clients. Anonymous strangers are cut off entirely. - Public doorbell endpoints gained abuse-control rate limits (7.1 above) and voicemail per-resident/day caps (20/day). - Flutterwave webhook handlers switched to constant-time HMAC comparison (`hmac.compare_digest`) closing a theoretical timing side-channel. - CORS moved from wildcard `*` to an explicit allow-list of `contritrack.com` , `www.contritrack.com` , the Emergent deploy URL, and the preview URL.

7.4 Terms & Conditions Bumped to v5

Effective July 2, 2026. § 7C ("Scan2Call — Video Doorbell Service") added in full, covering visitor consent (mic/camera/GPS), resident controls, wallet-based per-minute billing with the 15-second refund window and ₦100 welcome bonus, physical QR sticker orders, and rate limits. § 4.4 sub-processor table extended with **ipwho.is**, **Cloudflare**, and **OpenStreetMap Nominatim** (restored for visitor GPS reverse-geocoding). § 6.4 pricing table extended with the ₦30/min call rate and the sticker line item. § 6.7 added covering wallet top-ups + refunds. § 15.5 captures the changelog. `TermsGate` storage key was bumped `terms_accepted_v4` → `v5` so every existing user re-consents before their next session touches the new Scan2Call flows.

7.5 Engineering — Restart-Safe Billing + Diagnostics

- **Restart-safe billing watchdog.** Each active per-minute debit now stamps `last_tick_at` on the ring document. On FastAPI startup, `resume_active_billing_watchdogs()` sweeps every `status="answered"` ring, retro-bills every full 60-second window that elapsed during downtime (idempotent via unique `tx_ref = call_<ring_id>_min<N>`), auto-ends calls whose wallet drained mid-catch-up, and re-spawns the async billing task. Zero revenue leak on redeploys.
- **Agora error diagnostics.** The generic "Could not connect" toast was replaced with step-tagged, code-aware error messages ("Microphone blocked. Enable mic access…", "join (UID_CONFLICT): …", etc.)

and a `console.error` log line so DevTools carries the exact SDK code. Payload validation catches missing WS fields before touching Agora.

---

8. What's New in v6 — February 14, 2026

This release retires the Google Maps subscription paywall, removes OpenStreetMap as the user-facing default basemap, and ships dark mode as the new brand default. The net effect: every Guard user — citizens, fleet drivers, government dispatchers, public lookup visitors — now sees the same premium Google basemap and Safe Route AI Advisory the moment they open the app, with zero gating.

8.1 Google Maps Free for All

The ₦500,000/yr system-wide Google Maps subscription that existed in v4 has been retired. Google Maps tiles are now served unconditionally from `mt0-3.google.com` (the free raster endpoint Google has historically left open for embedded apps). Operators absorb the Google Cloud Platform billing directly via their own Google account, unmediated by a Flutterwave product line.

8.2 OpenStreetMap Demoted to a Silent Fallback

The OSM/CARTO basemap is no longer offered as a user-facing toggle. It survives only as `<TileLayer errorTileUrl={…}>` — when an individual Google tile request fails (rate limit, regional block, etc.) Leaflet quietly substitutes a CARTO dark/light tile so the user never sees a blank square. The "OSM | Google" tab control was removed from every map. Nominatim reverse-geocoding is also retired in favour of Google's Geocoding API.

8.3 Dark Mode as the Default Theme

First-time visitors and PWA installs now boot into **dark mode** regardless of OS preference. The brand-aligned colour palette matches the new shield icon and the rest of the marketing aesthetic. Users who explicitly toggle to light mode still have their choice persisted via `localStorage` and never see the default applied again.

8.4 Terms & Conditions Bumped to v3

Effective Feb 14, 2026. § 7C ("Google Maps Subscription — System-Wide Upgrade") was removed in full. § 4.3, § 4.4, § 6.1, § 6.3, and § 6.4 were edited to drop all references to OpenStreetMap and the retired subscription product. § 15.3 captures the changes; § 15.4 preserves the v2 changelog for the audit trail. `TermsGate` storage key was bumped `terms_accepted_v2` → `v3` so all existing users are re-prompted to accept the simplified Terms.

8.5 Engineering Cleanup

Five files were deleted from the React tree (`MapProviderContext` , `MapProviderTabs` , `SafeRouteLocked` , `useGmapVerify` , plus the related provider wiring) and the matching backend `routes/maps.py` was reduced to

a single compatibility stub returning `{active: true}` so any pre-cached service-worker bundle still polling `/maps/google/status` gets a permissive answer. The `donations.py` Flutterwave webhook handler no longer routes `guard_gmap_*` transaction prefixes through the maps finalizer — any in-flight gmap payments (vanishingly unlikely) fall through to `_finalize_donation` and are recorded as generic successful payments. SW cache bumped `guard-v40-shield-icon` → `guard-v41-google-only-dark-default`.

---

9. What's New in v5 — February 13, 2026

This release ships citizen-requested evidence richness, automated weekly compliance reports for administrators, and a deep engineering pass that hardened our biggest backend hotspots and tamed our largest frontend files. Every code change in this release was guarded by a regression test suite — Guard's automated backend tests grew from 0 to **132 passing tests** during this cycle.

7.1 Image Attachments on Alerts

Citizens can now attach up to **three (3) photographs** to any alert — for security, fire, accident, banditry, eCaution, location-share, or medical incidents — directly from the alert creation dialog or after-the-fact from the alert feed. Each image is encoded as Base64 (max 5 MB raw, JPEG/PNG/HEIC) and stored inside the alert document. To keep mobile data usage minimal, the **active-map feed transmits only the photo count, not the bytes** — viewers fetch images on demand via `GET /api/alerts/{id}/images` only when they expand a gallery or open the lightbox. Admins (and the alert creator) can add or replace photos at any time before the alert is resolved. Resolved alerts keep their photos for the audit retention window.

7.2 Weekly Digest Email for Administrators (Resend + APScheduler)

Verified administrators may opt in to a **Monday-06:00 (Africa/Lagos) compliance digest**, delivered via Resend. Each digest carries an inline HTML summary (alerts by type, top destinations, sanction count) and three CSV attachments (alerts, sanctions, ePass scans) for the previous 7 days. The CSVs use the same RFC-4180 + UTF-8-BOM format as the manual exports (so Excel renders ₦ and Yoruba diacritics correctly). Background scheduling is handled by `APScheduler` running inside the FastAPI process; the Monday timer fires regardless of which pod is currently serving traffic.

7.3 Digest Preview Modal + One-Tap Sidebar Subscribe

Admins can preview the exact HTML body of their weekly digest before subscribing — a sandboxed `<iframe srcDoc=⋯>` renders it 1:1 with what the email will contain (a `Send now` button exists for instant delivery). The preview is reachable from two places: the existing `/admin/activity` card and a new **compact Dashboard sidebar CTA** that collapses to a tiny `✓ WEEKLY DIGEST ON` pill once subscribed. The shared `DigestPreviewDialog` component is used by both surfaces so behavior stays identical.

7.4 Engineering: Frontend Modularisation

Four monoliths were broken into small, single-responsibility components — **no behavior changes, all `data-testid`s preserved** so existing UI tests run unmodified:

Component	Before	After	Extracted into
`Dashboard.js`	469	**164**	`pages/dashboard/` (LocationStatus, Desktop/Mobile layouts, 4 hooks)
`AlertFeed.js`	333	**72**	`components/alerts/` (AlertCard, EvidencePin, alertMeta)
`AlertImageGallery.js`	179	**132**	`components/alerts/` (Lightbox, Thumbs, UploadPanel)
`SafeRoutePanel.js`	495	**284**	`components/safe-route/` (Locked, Header, DestinationInput, AdvisoryCard, Options) + `hooks/useVoiceDestination` + `utils/saferoute`

7.5 Engineering: Backend Refactor + 132-Test Regression Suite

Two high-complexity backend hotspots were tamed by extracting pure helpers, each independently testable:

- `vote_on_alert` (was 98 lines, cyclomatic 24 — handles community-vote → sanction triggering with ₦10k fines) split into 5 helpers (validation, vote-record, count-recompute, trust adjustment, sanction trigger). Orchestrator now ~30 lines.
- `escalation_checker` background task (was 67 lines, nesting depth 5) split into 4 functions including a pure `_decide_escalation_level` decision function. Outer loop now ~10 lines.

A new regression test suite — **132 passing tests** across `test_vote_refactor_regression.py`, `test_escalation_refactor.py`, `test_epass_fleet.py`, `test_magic_link_auth.py`, `test_recovery_and_sanction.py`, `test_plates_in_motion.py` — guards every code path that touches sanctions, escalations, peer-voting, magic-link auth, recovery flows, ePass fleet bulk operations, and stolen-plate alerts. Session-scoped asyncio loop fix (in `pytest.ini` + `conftest.py`) keeps Motor's `AsyncIOMotorClient` healthy across the whole run.

7.6 Frontend Code-Review Pass

Array-index React keys replaced with stable composite keys in `AlertImagePicker`, `AlertImageGallery`, and `AdminActivityPage` recent-routing-calls. Type hints added to `core/geocode.py` and `routes/health.py`. The shared `DigestPreviewDialog` removed ~140 lines of duplicated Dialog+iframe code between the admin-activity page and the dashboard sidebar.

---

10. What's New in v4 — February 03, 2026

This release delivers ePass monetisation, Google Maps integration with AI traffic advisory, Claude-only OCR with telemetry, and a deeper safety-net for drivers.

7.1 Google Maps Layer + System-Wide Subscription (₦500,000/yr)

Every map in Guard now offers an **OSM | Google** tab toggle at the top-left. OSM stays free forever. Google's photorealistic basemap (with the rich Nigerian landmark coverage that OSM lacks — *"Murtala*

Muhammed International Airport", *"National Stadium Surulere"*, *"Tarkwa Bay Beach"*) unlocks system-wide via a **single ₦500,000-per-year Flutterwave subscription** that any system admin can pay. Once active, every Guard user gets the upgrade — citizens, fleet drivers, and government dispatchers all see the same richer map. System admins can also enable it for free as an internal toggle.

7.2 Safe Route AI Advisory (Driving Mode)

With Google Maps unlocked, Driving Mode gains a destination input and **AI-recommended route picker**. Backend calls Google Directions API with live traffic + alternatives, decodes each route's polyline, cross-references active Guard alerts (security / fire / banditry / accident / eCaution within 500 m of the path, last 6 h), and asks **Claude Sonnet 4.5** for the safest fast pick + a one-sentence advisory like *"Route via Eko Bridge is safest with no alerts and fastest at 18 minutes."* The chosen route renders as an emerald polyline on the map; the advisory is spoken via TTS in the driver's preferred Nigerian language (Yoruba, Hausa, Igbo, Pidgin, English).

7.3 Voice Destination

A microphone button next to the destination input lets the driver say *"Take me to Yaba"* hands-free. Browser Web Speech API recognises the phrase, strips natural-language preambles, and auto-fires Advise. Pulsing red "Listening…" indicator with clean error fallbacks.

7.4 Live Re-route + Arrival Detection

After a route is found, drivers can toggle **Live re-route**. A 60-second poll re-evaluates traffic + Guard alerts each tick. The system only toasts and re-speaks the advisory when the recommendation *meaningfully changes* (new alert appears on the chosen route, or an alternative becomes safer) — quiet otherwise so it doesn't nag. A LIVE pill shows the live distance-to-destination (e.g. *"LIVE 6.8 km"*). Once the driver dwells within **100 m of the destination for 30 s**, Live re-route auto-stops, the polyline is cleared, and *"You have arrived"* is spoken in the driver's language.

7.5 Route Cache (commute-hour cost savings)

Every successful route advisory is stored in a `routing_cache` collection keyed by `(origin_grid_~110m, destination, 5-min-bucket)`. Two drivers in the same neighbourhood asking for the same place within 5 minutes share a single cached response — second call returns in **~158 ms** at **$0 cost** vs ~3 s for a fresh Google + Claude call. Real-world impact: 30-70% hit rates during Lagos commute peaks, slashing the GCP bill proportionally.

7.6 Maps Usage Dashboard (admin)

A new **Maps Usage** tab under `/admin/activity` surfaces near-real-time spend control: Today / Last 7 days / Cache hit rate / Month-to-date / All time, top destinations, recent calls with `cache` badges and per-call latency, and a rough cost estimate (US$0.013/call × Google calls). Lets admins watch the GCP bill build up without leaving the platform.

7.7 ePass Annual Vehicle Subscription (₦5,000/yr)

Each registered vehicle now requires a ₦5,000/year subscription via Flutterwave to remain “active” on the network. Vehicles register in a `pending` state and become `active` on payment confirmation, with `expired` surfaced once the year elapses. The Vehicles table shows badges (UNPAID / ACTIVE-until-DATE / EXPIRED) and a one-click **Activate ₦5,000/yr** or **Renew** button. Renewals add 365 days from `max(now, current_expiry)`. Flutterwave webhook routes by transaction-reference prefix (`guard_veh_*`), shared with donation and Google-Maps subscription flows for unified accounting.

7.8 Claude-Only OCR Pipeline

The hybrid Tesseract + Claude fallback was retired. The cropped JPEG now goes straight to `/api/ocr/plate` where **Claude Sonnet 4.5** reads the whole image with Nigerian-format context (XXX-NNNXX / XXX-NNXX, position rules for letter vs digit slots, ignore federal banner / state crest / dealer plates). Output protocol is `PLATE|CONFIDENCE` (or `UNREADABLE|0`); reads below 70 confidence or that don’t match the Nigerian shape are rejected as clean misses so the UI prompts the user to retype or retry. Image hashes are cached so repeat scans of the same frame are free.

7.9 OCR Accuracy Telemetry

A new collection `ocr_logs` captures every `/api/ocr/plate` call (image hash, predicted plate, confidence, engine, format match, latency, accepted, user). The frontend posts a follow-up `/api/ocr/feedback` row when the user finally submits — capturing whether they kept the prediction or corrected it. Misses are deduped on a `ocr_misses` collection with reason + count. The ePass Analytics tab gains an **OCR Accuracy (Claude)** card showing total scans, accept rate, real user-kept rate, average latency + confidence, and a Recent Misses list. Tunable confidence floor (`MIN_CONFIDENCE = 70`) gives admins one dial to balance accuracy vs availability.

7.10 Mobile Header — Slide-in Sheet

The overflow-scroll mobile nav (which hid buttons until you scrolled) was replaced with a **slide-in** `Sheet` **menu** on screens `< md` . Top bar on mobile is now just: GUARD logo + Donate + Bell + Hamburger — clean and never crowded. Hamburger opens a side panel with profile/Trust header, all nav links (Dashboard, Reports, Intelligence, ePass, Users, Activity), theme toggle, Terms, Technical paper, system status pulse, and Sign-out pinned to the bottom.

7.11 Multi-Language UI i18n via TTS (Yoruba / Hausa / Igbo / Pidgin)

Alert announcements are translated on-the-fly via Claude Sonnet 4.5 and cached in a `translations` collection. The browser’s Web Speech API speaks the translated message — eCaution descriptions, security toasts, route advisories, arrival notifications. Fallback to English if the OS lacks the relevant voice pack.

7.12 Over-Speed Alerts in Driving Mode

Driving Mode now monitors the GPS-derived speed. When the driver crosses **120 km/h** (default, configurable per user), an overlay screen appears with the live speed and the over-speed warning is spoken via TTS. Critical for highway and night-driving safety.

7.13 Optional GPS on Public `/lookup`

The public stolen-vehicle lookup page no longer hard-fails when location is unavailable. Users without GPS (or who block it) can still type a plate and check the registry — the lookup just proceeds without proximity context.

7.14 Made-with-Emergent Badge Removed

The Emergent branding badge was removed from `index.html` for a cleaner production look.

---

11. What’s New in v3 — April 28, 2026

8.1 Live Video Reports (powered by Jitsi Meet)

Alert creators can now launch a **real-time video room** so administrators within proximity can witness an incident as it unfolds. Two-way audio + video, manual end, no server-side recording. Backend endpoints (`/api/live/*`) enforce a proximity gate so only users physically near the alert (or any admin) can join.

8.2 Driving Mode — Background eCaution Tracking

A windscreen-friendly toggle that activates a continuous high-accuracy GPS watch, requests a Screen Wake Lock so the device doesn’t sleep mid-trip, and announces nearby road cautions (potholes, checkpoints, flooded roads, etc.) via Text-to-Speech in the user’s chosen Nigerian language. Position is throttled-synced to the backend (every 30 s OR every 75 m moved) so proximity-based pushes stay fresh without burning battery.

8.3 “Live Now” Dashboard Banner

A pulsing red strip pinned to the top of the dashboard that lights up whenever any alert in the network has an open live-video room. Multi-room aware (a `+N more` dropdown lists every concurrent live alert), per-session dismissable, and WebSocket-driven so it appears and disappears in real time without polling.

8.4 Push Notification Action Buttons

Native action buttons render on every push notification: - **Admin:** `Resolve` + `Navigate` - **Citizen:** `I'm safe` + `Navigate`

`Navigate` opens Google Maps directions to the alert in a new tab. `Resolve` calls the resolve endpoint without ever opening the app. `I'm safe` records a `safe_check_in` row, broadcasts to admins, and notifies the alert creator that someone in proximity is unharmed — with one tap from the lock screen.

8.5 Push & Notification Deep-Linking

Tapping any push or in-app notification deep-links to `/dashboard?alert=<id>` which flies the map to the alert, auto-opens the popup, scrolls the matching feed card into view, and gracefully handles already-resolved alerts with a “no longer active” toast.

8.6 Admin CSV Exports

One-click CSV downloads for the four admin tables: - **Sanctions** (User ID, Name, Email, Reason, Sanctioned-At, Trust Score) - **Appeals** (filterable by status — pending / cleared / upheld / all) - **ePass scan logs** (Plate, Result, Denial Reason, Premise, GPS, Scanner, Time) - **Alerts** (existing — kept)

Output is RFC-4180 escaped and prefixed with a UTF-8 BOM so Excel renders ₦ and Yoruba diacritics correctly.

8.7 Reverse Geocoding for Alerts

Every alert is now decorated with a human-readable street address (e.g. “Marina, Lagos”) via OpenStreetMap Nominatim. Cached aggressively in MongoDB and applied asynchronously so alert creation latency is unaffected.

8.8 Trust & Appeals UX

The `/trust` page shows each user their score, the events that move it, and concrete next-steps. Citizens can submit appeals with their reasoning; admins review them in a dedicated `/admin/appeals` inbox with WebSocket-driven live updates and a Cleared / Upheld decision flow.

8.9 OCR Hardening

Tesseract.js plate scanning now applies Otsu thresholding + letterbox cropping in the browser before recognition, materially improving accuracy under uneven lighting. Stolen-vehicle alerts ship with a small thumbnail of the scan as inline evidence. *(Superseded in v4 by the Claude-only pipeline — see §7.8.)*

8.10 WebSocket Singleton

Every previous component opened its own socket. We refactored to a `WebSocketContext` provider with a single shared connection per browser tab and per-handler subscriptions — ~50 % drop in idle network usage on dashboards held open for hours.

---

12. What's Coming Next

- **SMS fallback** via Twilio for users without smartphones (rural outreach)
- **Multi-language UI strings (i18n)** — full Yoruba / Hausa / Igbo / Pidgin translations of buttons, labels, and toasts (not just spoken announcements)
- **Live-room viewer count badge** ("3 watching") on the LIVE NOW banner
- **Banditry heatmap overlay** — visualize attack patterns over 30 days
- **Fleet CSV-template download** — one click, sample CSV, easier onboarding
- **New-device-detected security email** — alerts users on unusual sign-ins
- **USSD short-code integration** — report emergencies without data
- **PWA install-prompt A/B test** — lift Android install rates on the 2nd visit
- **Commute-pattern memory** — Guard quietly learns each driver's regular destinations + times-of-day to surface a "Leave by 7:42 AM to arrive at work by 8:30 (12 min added by traffic)" suggestion the moment the app is opened.
- **Aerial Recon (live satellite/3D fly-over) for admins** — visualise an escalated incident from above without leaving the dashboard.

---

13. Call to Action

For citizens: Visit the app, sign in with your email, enable push notifications, set your proximity radii. You now have a safety net every neighbor can reinforce.

For organizations: Subscribe to ePass for your premises. Contact `alerts@contritrack.com` to register your fleet in bulk.

For government agencies: Email us for an onboarding session. Your verified `.gov.ng` account automatically unlocks admin capabilities.

For everyone: If Guard helps you, donate. Any amount helps keep the platform running and free for the next citizen who needs it.

Contact

Operator: Contrivances Engineering Ent. Ltd. **Email:** alerts@contritrack.com **Donation account:** GTBank 0035453902 — Contrivances Engineering Ent. Ltd. **App:** https://contritrack.com **Terms & Conditions:** https://contritrack.com/terms

Guard — because seconds matter.

Paper revision: v6 — February 14, 2026. Major changes since v5 (Feb 13, 2026): retired the ₦500,000/yr system-wide Google Maps subscription — Google Maps tiles + Geocoding + Directions API are now free

for every Guard user; demoted OpenStreetMap to a silent `errorTileUrl` *fallback (CARTO dark/light) and removed the “OSM | Google” tab from every map; set dark mode as the brand default for first-time visitors and PWA installs; Terms bumped to v3 (removed § 7C in full + updated § 4.3, § 4.4, § 6.1, § 6.3, § 6.4 to drop OSM and the subscription product); 5 files deleted from the React tree (* `MapProviderContext`, `MapProviderTabs`, `SafeRouteLocked`, `useGmapVerify`, *plus provider wiring);* `routes/maps.py` *reduced to a single compatibility stub;* `donations.py` *webhook no longer routes* `guard_gmap_*` *prefixes.*

Earlier revision: v5 — February 13, 2026. Major additions since v4 (Feb 03, 2026): photo evidence attachments on every alert type (up to 3 images, Base64, on-demand fetch); admin-opt-in weekly digest email every Monday 06:00 (Africa/Lagos) via Resend + APScheduler with 3 CSV attachments; HTML preview modal + Dashboard sidebar one-tap subscribe CTA; deep engineering refactor pass (Dashboard.js 469→164, AlertFeed.js 333→72, AlertImageGallery.js 179→132, SafeRoutePanel.js 495→284, backend vote_on_alert and escalation_checker split into 9 focused helpers); 132-test backend regression suite covering sanctions, escalations, peer-voting, magic-link auth, recovery flows, ePass fleet, stolen-plate alerts; shared `DigestPreviewDialog` *and React-key/type-hint hardening from a full code-review pass.*

Earlier revision: v4 — February 03, 2026 added system-wide Google Maps subscription (₦500,000/yr Flutterwave); Safe Route AI advisory powered by Google Directions + Claude Sonnet 4.5; voice destination via Web Speech API; Live re-route with arrival detection (auto-stops at ≤100 m / 30 s dwell); route-response cache keyed by `(origin_grid_~110m, destination, 5-min-bucket)` *with admin Maps Usage dashboard; ePass annual vehicle subscription (₦5,000/yr per vehicle); Claude-only OCR pipeline replacing the hybrid Tesseract fallback; OCR accuracy telemetry with confidence floor; mobile header redesigned as slide-in Sheet; multi-language alert TTS in Yoruba / Hausa / Igbo / Pidgin with Claude-translated cache; over-speed alerts in Driving Mode; optional GPS on the public /lookup page; Made-with-Emergent badge removed.*

Earlier revision: v3 — April 28, 2026 added Live Video Reports (Jitsi Meet) with proximity gate, Driving Mode background tracking with Wake Lock + TTS announcements, “Live Now” dashboard banner, role-aware push notification action buttons (Resolve / I’m safe / Navigate), notification deep-linking, admin CSV exports for sanctions / appeals / ePass scans, OpenStreetMap reverse geocoding for alert addresses, OCR Otsu-threshold hardening with evidence thumbnails, /trust user-facing trust page, /admin/appeals workflow, and WebSocketContext singleton refactor.