

Technical Overview

Architecture and stack for IT and security teams.

Version v3.2 | Last reviewed: 24 July 2026 | Owner: CaseFlow Automation Ltd

Architecture summary

TotalSettle is a React single-page application served from a CDN, backed by managed Postgres with row-level access controls and serverless Edge Functions for server-side logic. The database is hosted in AWS eu-central-1 (Frankfurt) and the edge functions run in AWS EEA regions, all within the European Economic Area. AI calls are made server-side through the Lovable AI gateway to Google Gemini.

Stack

- Frontend: React 18 with TypeScript, Vite, Tailwind CSS.
- Backend: Postgres with row-level security, hosted in AWS eu-central-1 (Frankfurt). Edge Functions (Deno runtime, AWS EEA regions) for server logic.
- Storage: encrypted object storage for org logos and uploaded evidence.
- Email: Resend via Edge Function, with idempotency keys to prevent duplicates.
- SMS: FireText API (UK provider, UK-only routing) via Edge Function, with per-organisation sender IDs and delivery receipts via webhook.
- AI: Google Gemini accessed through the Lovable AI gateway. No provider retention. No training.
- Observability: structured logs and event tables for handler and customer actions.

Tenant isolation

Every public-schema table that holds operator data has row-level security enabled. Policies scope reads and writes to rows owned by the requesting organisation, enforced server-side via the database, not the front-end. Helper functions used by policies are marked SECURITY DEFINER and live in the public schema with a fixed search path. Roles are stored in a separate user_roles table, never on the profile, to prevent privilege escalation.

Authentication

- Operator users authenticate via email and password, with optional Google OAuth.
- Password reset and team invitation flows are handled by dedicated Edge Functions.
- Customer-facing settlement portal does not use accounts. Access is verified by case token plus vehicle registration plus postcode.
- Multi-factor authentication available for operator accounts.

Engineer report processing

- PDFs are parsed in the browser using PDF.js where possible.
- Extracted text is passed to the extract-engineer-report Edge Function.

- The function calls a shared piiRedactor module that substitutes opaque placeholders.
- The redacted text is sent to Gemini with a structured output prompt.
- The response is passed through piiReinserter to restore original values locally.
- If text extraction fails (image-only PDF), the original PDF is sent to Gemini as a fallback and the event is logged per case.
- Per-case redaction summary (counts by type) is written to the user_events table.

Settlement link verification

Settlement links carry a high-entropy token. On open, the customer enters vehicle registration and postcode. Both are compared server-side against the case. Mismatches return a generic verification failure with throttled retries. The token is nulled when the case's personal data is purged, so links are disabled automatically once the case completes or expires.

Data lifecycle and purge machinery

- On completion (claimant accepts), notify-claim-response triggers purge-case-pii, which nulls all claimant identifiers on the case, deletes evidence rows and storage objects, redacts the email send log, and stamps pii_purged_at. The anonymised skeleton (status, decision, valuation figures, timestamps, reference) is kept for reporting.
- A nightly safety-net sweep (purge-case-pii-sweep) catches any completed case the event-driven purge missed.
- A daily abandoned-case sweep (purge-abandoned-cases-sweep) anonymises any case still pending or negotiating 30 days after creation and marks it expired.
- A daily warning job (warn-abandoned-cases) emails the operator's retention alert contacts 7 days before a case reaches the 30-day cliff. Warnings are tracked per case so they are never duplicated.
- Every purge writes a row to a dedicated audit table (case_pii_purge_audit), and an automated weekly health check verifies the purges are firing.
- Tenant offboarding uses a dedicated, audited tenant-deletion routine that wipes the organisation's data.

Edge Functions of note

- extract-engineer-report: PII redaction, AI extraction, reinsertion, validation.
- ingest-engineer-report: orchestrates case creation from an uploaded report.
- send-settlement-link, send-transactional-email: outbound mail via Resend.
- send-claim-sms: operator-triggered SMS of the settlement link via FireText (org-branded sender, logged with a reference); firetext-dlr: delivery-receipt webhook updating the send log. SMS log personal fields are cleared by the purge machinery.
- notify-claim-response: alerts the handler on accept or dispute, and triggers the completion purge.
- purge-case-pii, purge-case-pii-sweep, purge-abandoned-cases-sweep, warn-abandoned-cases: the data lifecycle machinery described above.
- invite-team-member, set-member-password: identity and onboarding flows.
- delete-tenant: audited whole-organisation offboarding wipe.

Backups and recovery

- Daily encrypted backups of the database.
- Point-in-time recovery within retention window. Purged personal data falls out of backups as they rotate through the recovery window.
- Disaster recovery procedure documented and reviewed annually.

CaseFlow Automation Ltd is the holding company that publishes TotalSettle. Registered in England and Wales. ICO registration: ZC013423. Registered office: 7-9 Macon Court, Crewe, CW1 6EA. This document forms part of the TotalSettle policy pack and is provided for information. It does not constitute legal advice. For the signed DPA, sub-processor list, or DPIA, contact info@caseflowautomation.co.uk.