LedgerScan

Security overview
Early-stage product · November 2026

LedgerScan gives each serialized unit a verifiable identity, scores every scan for counterfeit risk, and records state changes on a blockchain ledger. This page describes what is implemented today and what is not yet in place, so your security team can assess a pilot accurately.

IDENTITY & ACCESS

- **Keycloak (OpenID Connect)** handles staff sign-in for the LedgerScan realm, with brute-force protection enabled in the realm configuration.
- The API **verifies RS256 token signatures** against the realm's published signing keys (JWKS) and refetches keys when a new key ID appears.
- **Role-based access** on every protected API route. Separate roles for platform owner, admin, manufacturer, enterprise manager, hub operator and store operator.
- Manufacturer portal passwords are stored as **bcrypt hashes**.

PRODUCT RECORDS ON-CHAIN

- A **Solidity registry contract** (OpenZeppelin *Ownable*) stores each unit's ID, serial, manufacture date, expected destination, metadata hash and state (minted, in market, active, flagged).
- Only the contract owner can register units or change their state; every change emits an event, giving an **append-only audit trail**.
- The contract holds **no consumer personal data**. Development deployments target EVM test networks (Polygon Amoy, Ethereum Sepolia).

SCAN RISK SCORING

- A separate **risk service** (Python, FastAPI) scores each scan on device registration, hub match, timing against the expected shipping window, and distance from the expected location.
- Each verdict returns its **reasons** (for example *DEVICE_UNAUTHORIZED*, *HUB_MISMATCH*), so a flag can be reviewed rather than taken on trust.
- Flagged units move to a **flagged state on-chain**; an unflag request workflow exists for review.

HOSTING & DATA

- Services run as **Docker containers** (API, risk service, Keycloak) behind a Traefik reverse proxy.
- Public endpoints serve **HTTPS with Let's Encrypt certificates**; TLS terminates at the proxy.
- Application data lives in **Supabase (PostgreSQL)**; Keycloak uses its own PostgreSQL database. A local SQLite store supports duplicate-scan detection.
- Consumers verify in a **mobile browser**; no app install is required.

NOT IN PLACE TODAY · ON THE ROADMAP

- **No certifications or audits yet.** LedgerScan holds no SOC 2, ISO 27001 or similar attestation, has not had a third-party penetration test, and is not validated against DSCSA, EU FMD or GAMP 5.
- **Independent multi-region checkpoints** (several nodes agreeing before a result is final) are designed but currently simulated in a single service.
- **Database-level tenant isolation**, published uptime commitments (SLA) and documented encryption-at-rest controls are not yet in place.
- **Secrets hardening** (removing development fallbacks, managed key storage for signing keys) is planned before production use.

Questions or a security questionnaire? Use the contact form at led-scan.com/#demo-cta

Prepared from the LedgerScan codebase, November 2026.