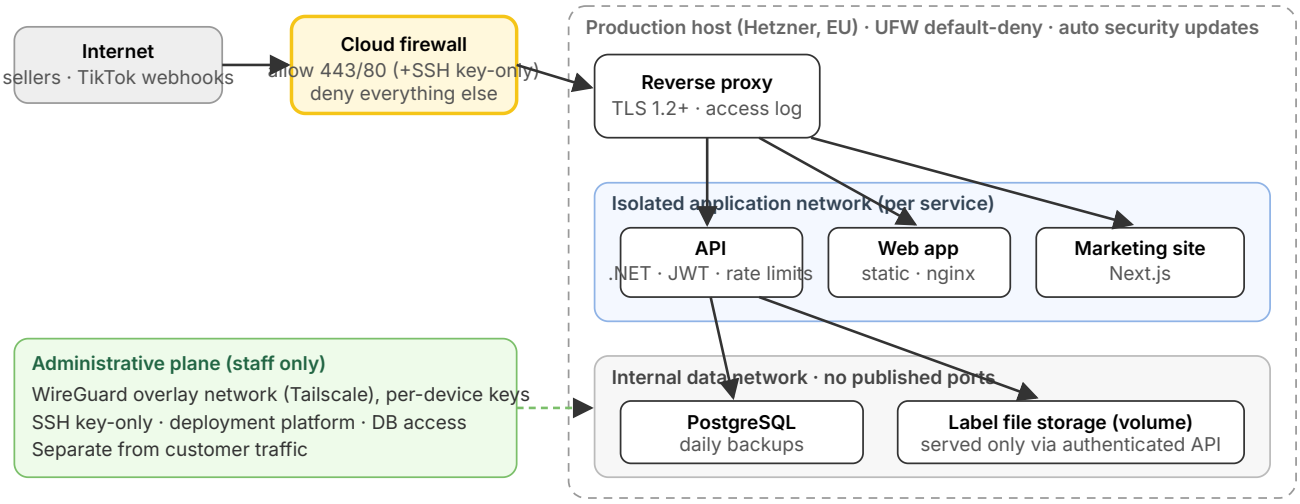
**Scan2Tray**

# Network Security & Segregation Overview

**Scan2Tray Ltd** (Company No. 17096824) · Version 1.0 · 24 September 2026 · Owner: Paul Hinett, Director · hello@scan2tray.com

Supplement to the Scan2Tray Information Security Policy, section 6.

## 1. Architecture




## 2. Segregation controls

| Boundary | Control |
|---|---|
| Internet → host | Cloud firewall (provider level) permits only HTTPS/HTTP ingress, SSH and the remote-support relay ports. Database, container management, metrics dashboards and all other services are blocked from the internet. Verified by external port scan. |
| Host | UFW host firewall with a default-deny inbound policy; SSH accepts key-based authentication only (password authentication disabled); operating-system security updates are applied automatically. |
| Ingress → applications | A single reverse proxy terminates TLS 1.2+ and routes by hostname. Application containers publish no ports of their own and are reachable only through the proxy. |
| Application → application | Each service runs in its own container on an isolated Docker network and can only reach peers it is explicitly attached to. |
| Application → data | PostgreSQL and label-file storage sit on an internal network with no published port; only the API containers are attached. |
| Staff → infrastructure | Administration happens over a private WireGuard overlay (Tailscale) with per-device keys, separate from customer traffic. Access is granted on a least-privilege basis and removed when no longer needed. |

## 3. Monitoring and threat prevention

- **Access logging.** The reverse proxy writes an access log for every request (retained with hourly rotation); the API logs authentication failures, authorisation failures, rate-limit triggers, webhook signature outcomes and administrative actions.
- **Automated blocking.** Authentication and device-pairing endpoints are rate-limited per client IP (HTTP 429); abusive sources are blocked with per-IP firewall rules. Sustained abuse alerts the

security owner.

- **Health monitoring.** Database connection saturation and transaction leaks trigger email alerts to the security owner; application metrics are collected to an internal, non-public monitoring stack.
- **Webhook hardening.** Inbound marketplace webhooks are HMAC-verified against the raw body and de-duplicated; unsigned or invalid requests are rejected with HTTP 401 and never processed.
- **Patching.** OS security updates are unattended; application dependencies are pinned and updated regularly.

## 4. Review

This overview is reviewed with the Information Security Policy at least annually and whenever the hosting architecture changes. Questions: hello@scan2tray.com.