Kitchen Toolkit Security Compliance Packet

Security questionnaire, data handling practices, retention policy, and pilot architecture

Last verified: 2026-06-06. Status: public pilot/procurement brief, not an audit report.

Important non-claims
No SOC 2 certification is claimed.
No ISO 27001 certification is claimed.
No HACCP certification is claimed for the SaaS, hardware, customer kitchen, or customer HACCP program.
No HIPAA compliance is claimed; the restaurant pilot is not intended to handle PHI.
No FERPA compliance is claimed; the restaurant pilot is not intended to handle education records.
No PCI-DSS compliance is claimed; the pilot is not intended to process payment-card data.
No production hardware certification is claimed; hardware remains pilot/pre-certification.

Executive Summary

- KT is designed for a contained restaurant pilot using operational kitchen data only.
- The pilot can run on KT-provided LTE with outbound HTTPS/TLS and no customer internal-network integrations.
- KT supports HACCP-style food-safety monitoring and recordkeeping, but does not claim HACCP certification.
- Manual food-safety procedures, calibrated reference checks, customer SOPs, and human decisions remain required.
- Formal SaaS/security, hardware, and food-safety workflow validation are separate roadmap tracks.

1. Security Questionnaire

1. What data does Kitchen Toolkit collect for the proposed pilot?

Operational kitchen records only: equipment IDs, device IDs, temperatures, timestamps, corrective actions, alerts, exports, support contacts, restaurant workflow content, device health, calibration records, and tenant settings needed to operate the service.

2. What data is intentionally out of scope?

The pilot is not intended to collect student records, HR/payroll data, finance/accounting records, payment card data, healthcare/PHI, government ID numbers, biometrics, camera surveillance, named employee performance records, emergency dispatch records, or consumer household data.

3. Does the pilot require internal customer network access?

No for the proposed contained pilot. KT can supply an LTE router/SIM so hubs use outbound HTTPS/TLS without UW Wi-Fi, VPN, AD/LDAP, SSO, student systems, HR/payroll, finance, payment, or healthcare system access.

4. How are admin users authenticated?

The admin dashboard uses password/session controls, signed cookies, CSRF protection, stale-cookie cleanup, tenant separation, and optional tenant email MFA. Broader SSO/RBAC/SCIM are roadmap items for larger deployments.

5. How are devices authenticated?

Provisioned devices send device IDs, payload hashes, HMAC signatures, and fresh nonces. The server verifies the stored device secret and rejects mismatches or replayed nonces before accepting telemetry.

6. How is data protected in transit and at rest?

Hub-to-cloud traffic uses HTTPS/TLS and signed payloads. Admin passwords are salted PBKDF2 hashes. Tenant secrets such as SMTP passwords, webhook secrets, alert settings, billing IDs, and BYO KV tokens are stored in an AES-256-GCM encrypted envelope and masked in admin reads.

7. Can customers export or delete records?

Yes. Admins can export retained logs and tenant data through product flows. Product-side delete flows remove live records from KT-targeted stores, but they do not rewrite Git history or customer-managed provider backups outside KT control.

8. Does KT replace food-safety procedures?

No. KT is an assistive operational monitoring and recordkeeping tool. Manual food-safety procedures, calibrated reference checks, customer SOPs, and human decisions remain in place.

9. Is KT HACCP certified?

No. KT supports HACCP-style food-safety recordkeeping and monitoring workflows, but does not claim HACCP certification for the software, hardware, customer kitchen, or customer HACCP program.

10. What certification work belongs on the roadmap?

SaaS/security: security assessment, penetration testing, IAM/RBAC/SSO/SCIM where needed, backup/DR, then SOC 2/ISO 27001 readiness when evidence matures. Hardware: FCC/ISED, applicable electrical/safety certification, and documented sensor validation/calibration. Food-safety workflow: continued HACCP-style workflow support and workflow validation where appropriate.

2. Data Handling Practices

- Purpose limitation: use pilot data to provide temperature monitoring, logging, alerts, exports, support, tenant administration, and food-safety workflow evidence.
- Data minimization: avoid sensitive personal data in free-text notes and keep the pilot scoped to kitchen operations.
- Tenant separation: records are keyed by tenant/device/equipment prefixes; non-default tenant bundle access is protected by middleware and short-lived content access flows.
- Storage model: default storage uses Vercel-hosted Next APIs and Upstash KV; logs/equipment can be routed to tenant-provided KV when configured.
- Access model: crew screens support operational logging; admin screens control exports, settings, equipment, retention, and telemetry health; platform support access should be limited to operational need.
- Credential handling: admin passwords are hashed, tenant secrets are encrypted, and masked values are returned to the admin UI.
- Third parties: Vercel, Upstash, GitHub, customer-selected SMTP/webhook providers, and KT-managed email infrastructure may process data needed to run the service.
- No sale of data: KT does not sell customer temperature data or operational records.
- Food-safety scope: KT supports HACCP-style monitoring and recordkeeping; customers remain responsible for regulatory compliance and decisions to hold, discard, recall, or serve product.

3. Retention Policy

Retention is implemented through KV TTLs, tenant/environment settings, product export tools, and product-side delete flows. Customers should export records before configured retention windows lapse when longer audit history is required.

- Equipment telemetry: default 180 days; configured by LOG_RETENTION_DAYS or tenant log_retention_days; key pattern log:equipment:{device}:{equipment}:{timestamp}.
- Manual logs: persist until deleted unless MANUAL_LOG_RETENTION_DAYS is set; key pattern log:{restaurantId}:{type}:{deviceId}:{timestamp}.
- Alert states: default 14 days; configured by ALERT_TTL_DAYS; includes battery, temperature, stale node, and stale hub alert state keys.
- Node presence: default 45 days; configured by NODE_LAST_SEEN_TTL_DAYS; key pattern node:lastSeen:*.
- Node last-value snapshots: persist until node/equipment detach or delete; not currently configurable; key pattern node:snapshot:*.
- Stale alert trackers: default 30 days; configured by STALE_ALERT_TTL_DAYS; key pattern alert:stale:*.
- Hub heartbeats and health summaries: persist until deleted unless HUB_LAST_SEEN_TTL_DAYS is set; key patterns hub:lastSeen:{deviceId} and hub:health:summary:*.
- Push subscriptions: approximately 90 days with pruning; key patterns pushsub:{endpoint}, pushsub:index, pushsub:byDevice:*, pushsub:byRestaurant:*.
- Restaurant bundles and docs: platform-managed Git-backed history; live files can be removed, but Git history is not rewritten by product delete flows.

Deletion limits
Product-side deletion targets live records in KT-controlled or tenant-configured stores.
Delete flows do not rewrite Git history or customer-managed provider backups, archives, or logs outside KT control.

4. Pilot Architecture

- Field layer: sensor nodes are wall-mounted away from prep areas, food, and food-contact surfaces. They monitor fridge/freezer air temperature; probes are suspended in open air and are not intended to touch food or other surfaces.
- Hub layer: LoRa nodes report to a KT hub. The proposed pilot can use a KT-paid LTE router/SIM. Hub traffic is outbound-only HTTPS/TLS to KT APIs, with signed payloads and nonces.
- Application layer: crew PWA supports offline-first logging and queues work for later sync. Admin dashboard supports review, exports, equipment records, tenant settings, node status, retention controls, and telemetry health.
- Cloud layer: Next/Vercel APIs receive logs, device heartbeats, exports, alerts, tenant settings, and support flows. Upstash KV stores default tenant-scoped telemetry, device auth, env/profile, alert, and health records.
- Optional customer storage: customers can configure tenant-owned KV for logs/equipment storage while platform device auth and tenant admin metadata remain operator-managed unless separately agreed.
- Document/content layer: restaurant bundles are GitHub-backed and mirrored into precache folders for offline installs; non-default tenant access is protected by middleware/content access flows.
- Pilot boundary: no customer internal Wi-Fi, VPN, AD/LDAP, SSO, student systems, HR/payroll, finance, payment, or healthcare system access is required for the contained restaurant pilot.

5. Roadmap Separation

- SaaS/security: security assessment, external penetration testing, IAM/RBAC/SSO/SCIM if needed, backup/DR, SOC 2 readiness, and ISO 27001 readiness.
- Hardware: FCC/ISED, applicable electrical/safety certification, production labeling, documented sensor validation, and calibration procedures.
- Food-safety workflow: continued support for HACCP-style food-safety workflows, mapped records/exports, temperature accuracy validation, calibration evidence, and workflow validation where appropriate.
- Do not describe the food-safety workflow track as product HACCP certification.

6. Source References

This packet is a public-facing summary generated from repository evidence and the compliance page. For deeper review, use the following source files and linked docs:

- pages/security-compliance.js
- docs/reference/security-and-data.md
- docs/reference/data-and-privacy.md
- docs/reference/retention-policy.md
- docs/reference/where-your-data-lives.md
- pages/admin/settings.js
- pages/admin/index.js
- pages/api/auth.js
- pages/api/csrf.js
- lib/auth/session.js
- lib/auth/device.js
- lib/tenant/env.js
- lib/tenant/storage.js
- pages/api/uploadEquipmentLog.js

- pages/api/uploadLog.js
- pages/api/hubHeartbeat.js
- pages/api/exportTenantData.js
- pages/api/deleteTenantData.js
- pages/api/buildRestaurant.js
- public/firmware/app_hub_wifi.ino