CrossConnect
CONNECT. MONITOR. OPTIMIZE.
BY Cybr IQ

Security & Architecture Reference

A plain account of how CrossConnect discovers, stores, protects, and audits your network data, with enough technical detail to stand up to review. Written for the IT and security professionals who must approve it before it touches their network.

Audience: IT architecture, network security, GRC, vendor risk

Scope: data collection, storage, encryption, identity, keys, logging, AI, compliance posture

Posture: read-only collection · self-hosted · advisory AI · tamper-evident audit

Document: security reference, 21 June 2026

Contact: contact_us@cybriq.io

0 How to read this document

This reference is built to answer the questions a security review asks, before you ask them. Each section names the actual mechanism, not just an adjective: the algorithm, the protocol, the port, the library and version, the exact behavior. Where a control is something you turn on at install rather than a default, it is labelled as such.

GA shipped & on by default CONFIGURABLE shipped, operator-enabled

DEPLOYMENT OPTION supported integration / hardening you select at install

1 Posture at a glance

CrossConnect is a network source-of-truth and operational-intelligence platform. It builds one model of your network, the devices, IP space, VLANs, cabling, circuits, routing, and the live state behind them, and lets your team ask it questions in plain language. Four facts frame every control in this document:

Read-only by construction

Discovery uses read-only credentials and read-only protocol operations. CrossConnect never writes configuration to a network device. No packet capture, no payload inspection, it reads switch-derived signals and the announcements gear already broadcasts.

Runs on your infrastructure

Self-hosted in your data center, private cloud, or a managed instance you control. Your network data stays in your PostgreSQL system of record. There is no mandatory vendor cloud in the data path.

Encrypted, with managed keys

Secrets are encrypted with AES-256-GCM under an envelope scheme: a key-encryption key (KEK) wraps a data-encryption key (DEK). The master key never ships in code. It is read from your environment, secret manager, or KMS (key management service), and it rotates without re-encrypting data.

Tamper-evident audit

Every change is written to a hash-chained, HMAC-signed audit trail. Altering any historical entry breaks the chain and is detectable on verification. The chain is the integrity evidence behind every answer.

The advisory rule, stated once. The conversational AI reads from the source of truth; it does not act on its own. It is scoped to the tenant and aware of the user's role. Every answer cites the record it used or says “I don’t know,” and any change it proposes is queued for a human to confirm before it commits. The AI never executes a change on a device, and never silently alters a record.

2 System context & trust boundaries

CrossConnect is a single deployable application backed by PostgreSQL, plus an optional formal-analysis sidecar (a helper process for whole-network config analysis). It reaches into the customer network only with outbound, read-only connections. It accepts operator and API traffic only on its own inbound application port. The diagram marks every edge with its transport and direction.

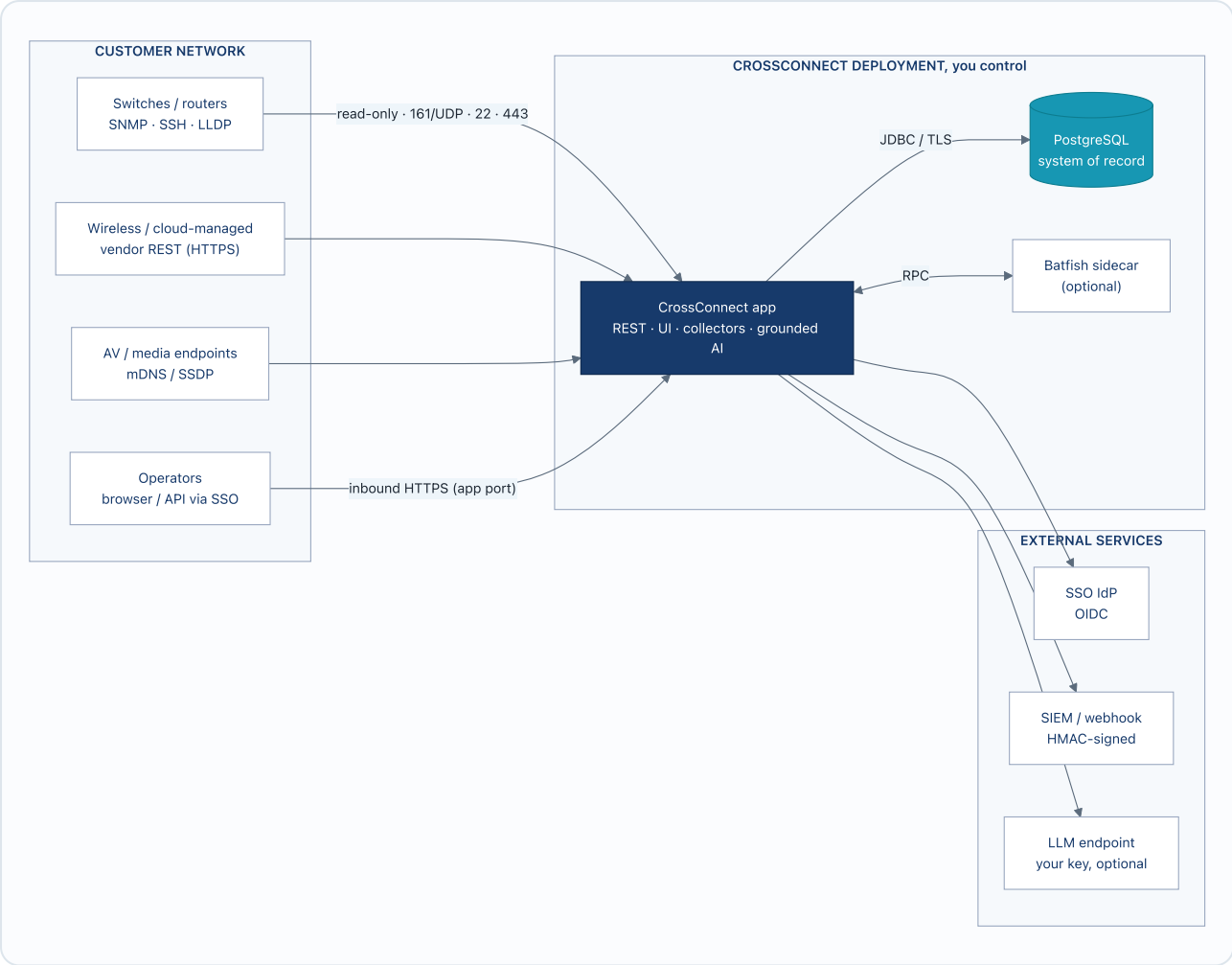

Figure 1. System context. Every connection into the customer network is outbound and read-only. Operator and API traffic is inbound to the application port only. The source of truth, the keys, and the data never leave the deployment you control.

Trust boundary	What lives there	How it is crossed
Customer network	Managed devices, wireless/cloud controllers, AV endpoints, operators	Outbound read-only discovery; inbound operator/API sessions to the app port
CrossConnect deployment	Application, PostgreSQL system of record, optional Batfish sidecar, encryption keys	Internal only; JDBC over TLS to Postgres, single-session RPC to Batfish
External services	Your SSO IdP, your SIEM/webhook receivers, an optional LLM endpoint	OIDC redirect, HMAC-signed outbound webhooks, AI calls under your own key

3 What we discover

CrossConnect collects network state and inventory facts. It does not collect user content, application payloads, or the contents of packets. Everything below is metadata about the network and the devices on it: facts the equipment already publishes through its control plane, read straight from there.

Category	Facts collected	Source
Device inventory	sysName, sysDescr, model, serial, software version, role, DRAM/flash, uptime, reachability	SNMP system & ENTITY MIB
Interfaces	name/alias/index, type, MTU, speed, MAC, admin/oper status, PoE draw, stacking	SNMP ifTable / POE-MIB
Layer-2 topology	LLDP neighbor adjacencies: remote chassis/port id, remote system name	SNMP LLDP-MIB
VLANs & VRFs	static VLAN ids + names, VRF names + route distinguishers, port assignments	SNMP Q-BRIDGE / MPLS-L3VPN MIB
IP address management	configured IPs + masks, prefixes, ARP/forwarding-derived endpoints (MAC→IP)	SNMP IP-MIB / BRIDGE-MIB
Routing	BGP sessions (local/peer AS, state), OSPF adjacencies (neighbor id, state)	SNMP BGP4 / OSPF MIB, Batfish
Multicast	IGMP querier state, group memberships per interface	SNMP IGMP-MIB
Precision timing	PTP domain, profile, grandmaster, clock class, port membership	SNMP PTPBASE-MIB
Environment	temperature, humidity, power-supply status sensors	SNMP ENTITY-SENSOR-MIB
Running configuration	full device config text for drift detection and formal analysis OPT-IN	SSH read of `show running-config`
Cloud-managed facts	VLANs, switch-port profiles, L3 firewall intent, SSIDs from vendor dashboards	Vendor REST API (HTTPS)
AV & media	service announcements (Dante/AES67/NDI/Q-SYS), media flows, vendor by MAC OUI	mDNS / WS-Discovery / SSDP
Wireless & occupancy	AP/SSID inventory, radio config, zone counts WHEN INTEGRATED	Wireless vendor cloud APIs
Operator & external input	documented records, bulk imports, asserted facts from other systems	UI / REST / inbound event API

> **What we explicitly do not collect:** packet payloads or the content of user traffic, no SPAN or mirror feeds, no deep packet inspection, no agents installed on user machines, and no personal data beyond the operator accounts you set up for the platform itself.

4 How we discover it, mechanisms & libraries

Every collection path is a real protocol implementation with a fast read timeout and a clean way to fail. A device it cannot reach is logged and counted, it does not block the rest of the sweep, and it never falls back to writing anything. The exact libraries and pinned versions are listed below so you can match them against your own CVE (vulnerability) feeds.

Mechanism	What it does	Library (pinned)	Default
SNMP v1/v2c/v3	Read-only GET / GETBULK walks of the MIBs in §3. v3 USM supports authPriv (SHA/AES).	`snmp4j 3.8.2`	ON
SSH config read	Interactive shell, runs read-only `show`-class commands, captures config text only.	`sshj 0.38.0`	OPT-IN
Vendor REST	HTTPS pulls from cloud-managed dashboards; bearer token; redirects blocked; guarded against SSRF (server-side request forgery).	JDK `HttpClient`	ON CREDENTIAL
Formal analysis	Whole-fleet config model for reachability, ACL, and IPAM-conflict questions.	Batfish sidecar (RPC)	OPTIONAL
Link-local discovery	Passive listen for mDNS / WS-Discovery / SSDP service announcements.	JDK NIO multicast	OFF
Inbound event API	Other systems POST facts they assert; these are held as proposals and never applied automatically.	Spring REST	AUTHN-GATED

Platform component	Version	Role
Java	21 (LTS)	Runtime
Spring Boot	3.4.0	Application framework, REST, filters
PostgreSQL JDBC	42.7.4	System-of-record driver
Flyway	10.20.1	Versioned schema migrations
Vaadin Flow	24.5.7 (LTS)	Server-rendered operator UI
LangChain4j	0.36.2	AI orchestration (provider-pluggable)
OpenTelemetry	1.43.0	Tracing / observability export

5 The collector security model

Most security teams turn here first, because this is a system that holds credentials and reaches into the network. CrossConnect is built so the answer to every concern is structural (something the design makes impossible) rather than procedural (something we promise to do).

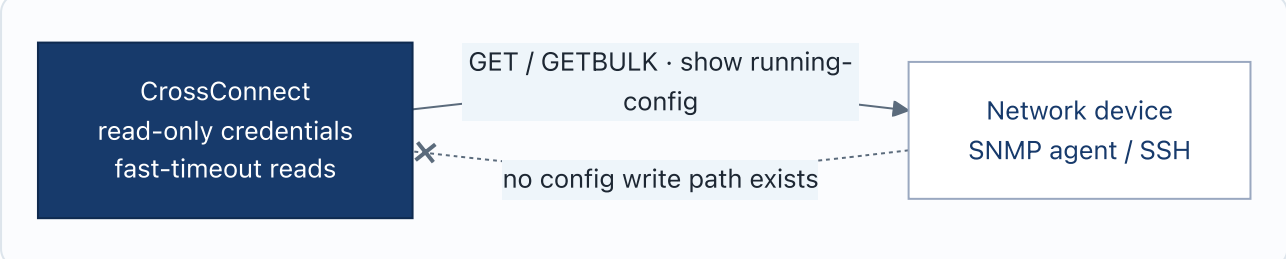

Figure 2. Read-only by construction. The product implements collection but not configuration. There is no code path that issues a `set`, a `configure`, or any state-changing command to a managed device.

Read-only credentials

Provision a dedicated, least-privilege service account. Read-only SNMP and a read-only SSH/show role are sufficient. The platform requests nothing more.

No configuration writes

The collector has no “apply” path to a device. It is collection-only software; remediation is advice for a human to act on, never an automated change.

Published command scope

The SNMP OID families and the read-only command set the SSH collector issues are enumerated in Appendix B, so you can scope and audit the account precisely.

Prefer the strong protocols

Use SNMPv3 USM (authPriv) instead of v1/v2c community strings, and SSH public keys instead of passwords. The platform supports the hardened option on every path.

Credentials stay encrypted

Device credentials are stored AES-256-GCM encrypted (§11). They are decrypted only in memory, only at the moment of a probe, and are never logged or repeated back in an answer.

No inbound exposure to the network

Discovery only ever connects outward from the platform. Optional collection on an isolated segment uses an outbound-only relay with no inbound listening ports.

6 Ports, protocols & egress

This is the complete list of connections. The only things that come in to the platform are the application port and the internal database port. Everything that touches the network goes outward. The full matrix is in Appendix A.

Direction	Port / protocol	Purpose	Auth
Outbound → devices	`161/UDP` SNMP	Read MIBs	v2c community / v3 USM
Outbound → devices	`22/TCP` SSH	Config read (opt-in)	key or password
Outbound → vendor cloud	`443/TCP` HTTPS	Cloud-managed facts	bearer API key
Inbound (app)	`8080/443 TCP` HTTPS	UI, REST	SSO session / JWT / API key
Internal	`5432/TCP` JDBC	System of record	DB credential, TLS
Internal	RPC to sidecar	Formal analysis	private network only
Outbound → you	`443/TCP` HTTPS	Webhooks, SIEM sinks	HMAC-SHA256 signed
Outbound → IdP/LLM	`443/TCP` HTTPS	SSO, optional AI	OIDC, your LLM key

7 Where data is stored

PostgreSQL is the single system of record. There is no secondary data lake, no analytics warehouse, and no copy held on the vendor side. Each kind of data is stored separately, so the protection can match how sensitive it is.

Data class	Stored as	Protection
Source of truth	Canonical entities: device, interface, cable, ip_address, prefix, vlan, vrf, circuit, network_service	Storage-level encryption (§10 L1); tenant-scoped
Observations (staging)	Append-only `discovered_*` rows, newest-per-key operative	Storage-level; auto-purged (§18)
Secrets	`snmp_credential`, `ssh_credential`, `secret`, cloud tokens, webhook secrets	Field-level AES-256-GCM (§10 L2)
Audit trail	`event_audit`, `system_event_audit`, `ai_audit_entry`	Hash-chained + HMAC-signed (§14)
Derived results	None persisted, compliance, quality, reachability are pure functions of a snapshot, short-TTL cached only	In-memory cache; no new facts written

Multi-tenancy. The tenant is the line that keeps data separate. Every row carries a `tenant_id`, every query filters on it, and foreign-key constraints stop any row from belonging to the wrong tenant. A request filter rejects any `/api/v1/*` call that does not resolve to an active tenant. A single-organization deployment simply runs as one tenant, and the same boundary governs instances that serve many organizations.

8 The observed-vs-documented trust gate

A discovered fact is never quietly accepted as truth. Observations land in a staging area first, where they are compared against the documented model and given a confidence score. They move into the source of truth only by a human action or an explicit rule that confirms them. This gate is the platform's central integrity boundary, and it is the reason an answer can be trusted.

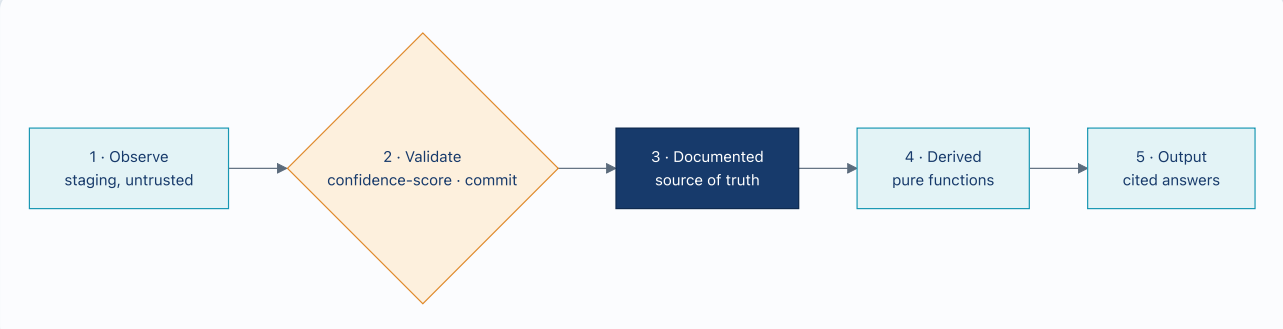

Figure 3. Data lifecycle. Confidence is earned by corroboration: two agreeing sources score *Confirmed*, a single source is *Inferred*, an observation resolving to no known entity is *Unconfirmed* and surfaced as a possible rogue. A commit is the only write path from observation to truth, and every one is audited.

9 Encryption in transit

- **Operator & API:** HTTPS/TLS terminates at the application (or your load balancer). HSTS is set on HTTPS responses; security headers (`X-Content-Type-Options`, `X-Frame-Options`) are applied by a response filter. GA
- **Database:** the JDBC connection to PostgreSQL supports TLS. A TLS compose overlay and managed-DB TLS are documented, and we recommend them for any database link that is not on the local loopback. CONFIGURABLE
- **Device collection:** SNMPv3 keeps the data private and unaltered on the wire (authPriv); SSH is encrypted by design; vendor APIs are HTTPS-only with redirects disabled. CONFIGURABLE
- **Outbound webhooks / SIEM:** sent over HTTPS, and the payloads are also HMAC-SHA256 signed, so the receiver can confirm they are genuine even apart from the transport (§13). GA

10 Encryption at rest & key management

Encryption at rest works in layers. Storage-level encryption covers all data. On top of that, field-level envelope encryption protects secrets with managed keys you can rotate. The underlying algorithm is AES-256-GCM (AEAD), with a fresh 12-byte random IV (initialization vector) per value and a 128-bit authentication tag. If a value has been tampered with, the tag fails on decrypt, so the change is caught rather than silently accepted.

Layer	Covers	Mechanism
L1 · Storage	All operational, audit, and inventory data	Encrypted volume / managed-DB encryption (your KMS or full-disk encryption)
L2 · Field	All secrets (device creds, API tokens, webhook secrets, AI keys)	AES-256-GCM via an application cipher, keyed by a per-deployment data key
L3 · Keys	The keys that protect L2	Envelope encryption: the KEK (key-encryption key) wraps the DEK (data-encryption key); the KEK is read from env / command / KMS

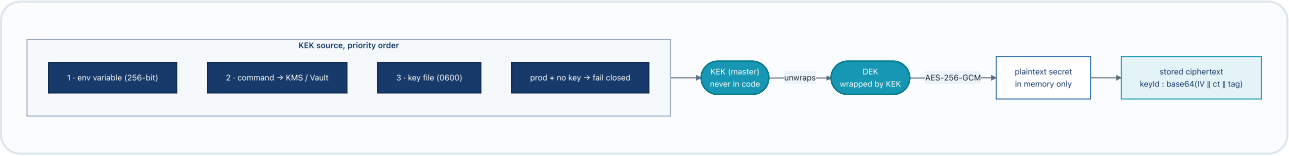

Figure 4. Envelope encryption. Secrets are sealed by a data key (the DEK). The DEK is in turn wrapped by a master key (the KEK), which never ships in code. Each stored value records which key version sealed it, so rotating the KEK only re-wraps the DEK, in milliseconds, and never has to touch the stored rows.

Key resolution (no weak defaults)

The master key is looked up in priority order: (1) an explicit environment variable, (2) an external command (a hook with a 15-second time limit that can fetch the key from HashiCorp Vault, AWS or GCP KMS, or a secret manager) DEPLOYMENT OPTION, (3) an auto-generated key file created with `0600` permissions on first boot. If a production profile starts with no key configured, the application **fails closed and refuses to start**, and a placeholder sentinel value always fails closed too. No encryption key is ever compiled into the build.

Key rotation

- **KEK rotation (routine):** a pre-flight check confirms the current KEK can decrypt every data key, the prior wrapped form is snapshotted so it can be rolled back, each DEK is re-wrapped under the new KEK with a round-trip check, and the event is written to the audit chain. No row ciphertext is touched. GA
- **DEK rotation (only on a suspected compromise):** a new data key is issued and the secret fields are re-encrypted under it, row by row, in a way that is atomic and can resume if interrupted. Because each value records its key version, the old and new forms can coexist while the migration runs. CONFIGURABLE
- **Restore:** rotation is reversible from the pre-rotation snapshot; the prior KEK is retired only after the new key is confirmed in service. GA

> **An honest caveat, stated up front:** rotating a KEK changes the master key and re-seals the data keys, but it does not by itself re-encrypt the historical row data. That is exactly why a separate DEK-rotation path exists, for the rare case of a suspected data-key compromise. Bring-your-own-key (BYOK) using your own KMS is supported through the key-command hook.

11 Secrets & credential handling

Secret	At rest	In use
SNMP community / v3 keys	AES-256-GCM, per-tenant	Decrypted in memory at probe time; never logged
SSH password / private key	AES-256-GCM	Decrypted only for a read session
Vendor cloud API tokens	AES-256-GCM	Decrypted for an HTTPS pull; SSRF-validated target
Webhook signing secret	AES-256-GCM	Used to HMAC outbound payloads; rotatable
AI provider key	AES-256-GCM (or env)	Loaded for the model call only; never echoed

No secret is stored in plaintext, written to application logs, or returned in an API response or AI answer. If you prefer, credentials can be pulled at runtime from your secret manager instead of the application database, using the same key-command mechanism described in §10. The platform admin secret is stored as a SHA-256 hash and compared in constant time (so timing cannot leak it); the token-signing secret is held as the HMAC key that signs sessions and the audit chain, and session signatures are likewise verified in constant time.

12 Identity, SSO, grouping & RBAC

Your identity provider says who the user is, their role says what they can do, and every request is scoped to the tenant. When SSO (single sign-on) is enabled, operators get no local passwords at all. The IdP (identity provider) stays the system of record for identity and group membership.

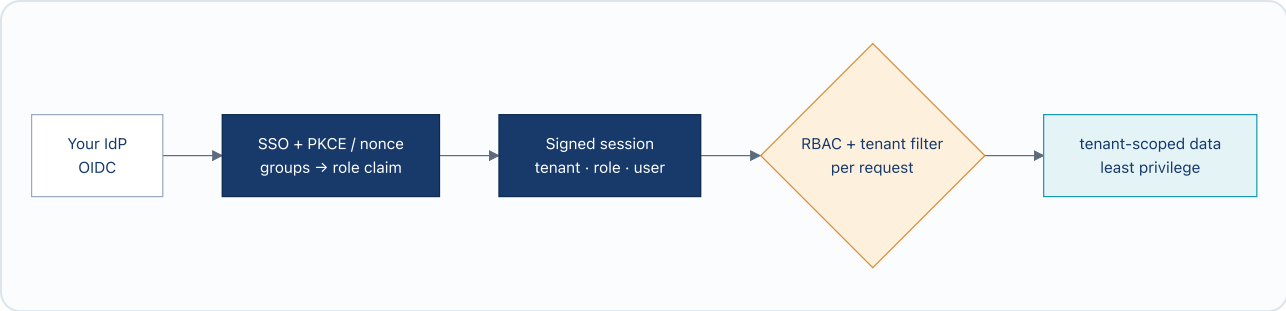

Figure 5. From identity to authorization. The IdP confirms who the user is and states their group membership. CrossConnect maps that group or role claim, issues a signed session tied to the tenant and role, and enforces RBAC (role-based access control) plus tenant isolation on every API request.

Authentication

- **SSO via OIDC** behind one enterprise-connection model, so any standard OIDC identity provider is supported. The OIDC path uses the authorization-code flow with PKCE (S256) and validates `iss`, `aud`, `exp`, and `nonce` (which defends against replay attacks). Group and role claims map to platform roles. CONFIGURABLE
- **Sessions are signed** (HMAC-SHA256), carry tenant, role, and user, and expire on a configurable TTL (default 8 hours). Signature verification is constant-time. GA
- **IdP-governed access:** when SSO is enabled, operators hold no local password and a session is minted only after a successful IdP sign-in, so revoking a user (or their group) in your directory stops them obtaining a new session here. The IdP stays the system of record for identity and group membership. CONFIGURABLE

Authorization (RBAC + grouping)

Role	Grants
Viewer	Read-only across inventory, compliance, and secrets metadata
Editor	Viewer + create/update/delete records, config and IPAM writes, report management
Admin	All of the above + user management, key rotation, and administrative endpoints

Roles are rank-ordered, meaning an action requires at least a certain role, and request filters enforce this: any request that changes data requires Editor or above, user management requires Admin, and administrative endpoints sit behind a separate admin credential that can be set to fail closed if it is not configured. IdP groups map onto these roles, so access is governed centrally by the groups your directory already maintains. Every `/api/v1/*` request is also tied to an active tenant, and an optional hardening flag rejects any request that does not present a signed session or API key.

13 API keys & webhooks

API keys

- **Issuance & scope:** keys are tied to a tenant and a role, so each key carries only the access that role allows. Inbound integration endpoints (presence, telemetry, admin operations) require a properly issued key, not just a guessable identifier. CONFIGURABLE

- **Monitoring:** key use, failures, and administrative actions are written to the audit trail with NTP timestamps, so issuance and use are reviewable. GA
- **Rotation & revocation:** keys carry expiry and revocation state and can be rotated without downtime by issuing a replacement and disabling the prior key. CONFIGURABLE

Webhook signing & rotation

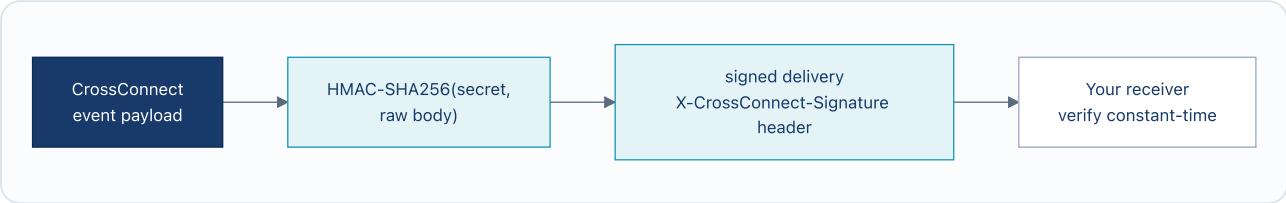

Figure 6. Proving a webhook is genuine. Outbound events are HMAC-SHA256 signed over the raw body and carried in an `X-CrossConnect-Signature` header, so the receiver can confirm they are genuine even apart from TLS, and you can rotate the shared secret with an overlap period (so there is no downtime).

The signing secret is stored encrypted, and each delivery carries the HMAC-SHA256 signature of the raw body in an `X-CrossConnect-Signature` header. On the receiving side, verification should use a constant-time comparison. Inbound webhooks the platform accepts (for example presence updates or facts asserted by another system) are themselves authenticated and held as proposals. They are never applied to the source of truth without passing the trust gate in §8.

14 Logging & tamper-evident audit

Every meaningful action, a record change, a config apply, a secret edit, a discovery run, a key rotation, an AI prompt, is captured in an audit trail. That trail is not just append-only: each entry is cryptographically linked to the one before it, so quiet tampering can be detected, not merely discouraged.

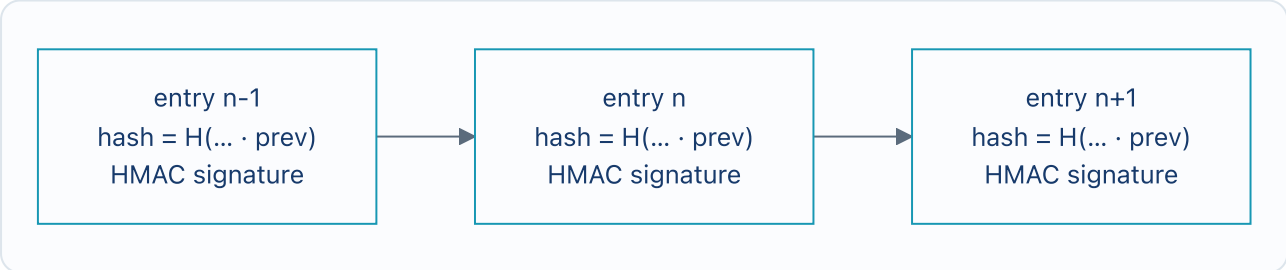

Figure 7. A hash-chained audit. Each entry's content hash includes the previous entry's hash, and the entry is HMAC-signed: `contentHash = SHA-256(tenant · kind · time · actor · payload · previousHash)`. Change any past record and every link after it breaks. A verification pass recomputes the chain and reports exactly where the break is.

- **What is recorded:** actor, action kind, timestamp, and a full before/after payload for each change; system events (discovery runs, key rotations) on a parallel global chain; every AI prompt, retrieval, and output on its own entry.
- **Integrity:** SHA-256 hash-chaining plus HMAC-SHA256 signing under the deployment signing secret. A verification routine walks the chain, recomputes the hashes, checks each link, and verifies the signatures in constant time. The result is a simple pass or fail you can show on screen or export as compliance evidence.

- **Export & monitoring:** tamper-evidence only helps if someone is watching, so the audit stream is meant to be shipped to your SIEM and monitored there. Outbound signed sinks and OpenTelemetry export are built to support that.
- **Retention:** chain-aware retention trims past the configured window while preserving link integrity (§18).

15 AI / LLM data handling

The assistant is grounded and tightly constrained. It answers questions about the source of truth, and it does not touch the network.

Grounded & cited

Every answer cites the record it used or says "I don't know." A citation validator rejects any answer that points to a record the tools did not actually return, so the model cannot invent devices, IPs, or relationships.

Tenant- & role-scoped

The assistant sees only the records the user is allowed to see. It honors RBAC and tenant isolation exactly as the rest of the API does.

Advisory only

It explains, highlights, and tells you how to fix. Any change is queued as an intent for confirm-before-commit by a human. It never executes a change on a device.

Fully audited

Every prompt, the records retrieved, and the output are logged to a tenant-scoped AI audit entry, so any answer can be reconstructed and reviewed.

Provider & data flow. You choose the AI provider. It can run against a model endpoint you name, under your own API key, or you can switch it off entirely, in which case the platform falls back to fixed, non-AI responses. Only the question and the specific records needed to answer it are sent to the model. Secrets and ciphertext are never included. If data residency for AI matters to you, point the assistant at a private or in-region model endpoint. CONFIGURABLE

16 Application security & SDLC

- **Static analysis & dependency scanning:** SpotBugs (which inspects the compiled bytecode for bugs) and OWASP Dependency-Check (which scans dependencies for known CVEs against the NVD) both run in the build and produce reports.
- **Test discipline:** we write property and invariant tests, not just example tests. Security-specific suites cover the JWT codec, admin-secret handling, RBAC enforcement, tenant isolation, the audit hash-chain math, encryption round-trips, and regressions in known CVE classes, all exercised against a real PostgreSQL in integration tests.
- **Recent hardening (verified, in the codebase):** the server-side request forgery (SSRF) risk on vendor-URL fetches is closed (HTTPS public targets only, no redirects, and RFC1918/loopback/metadata/CGN/ULA addresses rejected); a race condition in key material is fixed; a denial-of-service in the config parser is now bounded; webhook and admin endpoints no longer trust a bare identifier and require an issued credential; and UI actions that change data are gated by role.

- **Secure-by-default options:** opt-in flags make the deployment fail closed when the admin secret, the signing secret, or request credentials are missing, so a production misconfiguration is loud rather than silent.
- **No secrets in code, no TODO placeholders in production paths** are enforced engineering rules; keys and credentials come only from the environment, a key file, or your secret manager.

17 Vulnerability management & disclosure

- **Dependency posture:** pinned versions (§4) mapped against OWASP Dependency-Check; updates are driven by CVE disclosure and the scan output.
- **Reporting a vulnerability:** security issues can be reported to contact_us@cybriq.io; a coordinated-disclosure process and acknowledgement are provided on receipt.
- **Patch handling:** because you host the platform yourself, security updates arrive as new builds you roll out on your own schedule. Releases that close a security finding come with advisory notes.

18 Retention, deletion & residency

Data	Default retention	Mechanism
Observations (`discovered_*`)	rolling window (operator-set)	Scheduled staging purge sweep
Audit chain	rolling window, integrity-preserving	Chain-aware purge sweep
Health / sensor samples	latest-per-sensor + window	Retention sweep
Source of truth	lifecycle of the record	Soft-delete (`deleted_at`), hard-delete on request
Occupancy / wireless	windowed	Retention service WHEN INTEGRATED

Residency. Because the deployment and its PostgreSQL live in the region and account you choose, data residency is whatever you set it to. No network data has to be processed on the vendor side. The only optional external data path is an AI model endpoint, which you pick (you can keep it in-region or private) or turn off.

19 Deployment & operational hardening

CrossConnect ships as a set of containers (the application, PostgreSQL, and the optional analysis sidecar) and runs under Docker Compose, Kubernetes, or a managed-database setup. The recommended production hardening checklist follows:

Identity & access

Set the admin and signing secrets (enable fail-closed validation); enable SSO; require a signed session or API key on the API; map IdP groups to roles.

Keys

Provide the master key from your KMS/secret manager or persist and back up the generated key file separately from database backups; schedule KEK rotation.

Transport & storage

Terminate TLS at the app or LB; enable PostgreSQL TLS; place the database volume on encrypted storage or a managed encrypted instance.

Collection

Use a dedicated read-only service account; prefer SNMPv3 authPriv and SSH keys; scope the account to Appendix B; keep config collection opt-in where not needed.

Observability

Export the audit stream to your SIEM and OpenTelemetry traces to your collector; monitor audit-chain verification and key-rotation events.

Network

Allow only the outbound discovery ports the deployment needs; the platform opens no inbound listeners toward the customer network.

20 Compliance posture & framework alignment

We describe where we stand in precise, defensible language. CrossConnect's controls line up with the recognized frameworks, and we claim no certification we do not actually hold.

Framework	Posture & phrasing
SOC 2	An attestation, not a certification. The control design (access, encryption, logging, change management) is aligned to the Trust Services Criteria, and a report is being pursued as the program matures.
ISO 27001	Controls aligned to Annex A; certification is a roadmap item, stated as "in progress" rather than achieved until an accredited audit completes.
NIST CSF 2.0 / 800-53	Aligned to (voluntary, not certifiable): Govern, Identify, Protect, Detect, Respond, Recover map to the controls in §§5–17.

Control-to-framework crosswalk (illustrative). Tenant isolation & RBAC → SOC 2 CC6 / ISO A.5.15 / NIST AC-2,AC-3. Encryption at rest & key management → CC6.1 / A.8.24 / SC-12,SC-28. Tamper-evident audit → CC7.2 / A.8.15 / AU-2,AU-9. SSO & provisioning → CC6.1 / A.5.16 / IA-2,IA-4. Vulnerability management → CC7.1 / A.8.8 / RA-5,SI-2. A full crosswalk and a pre-filled CAIQ/SIG-Lite response are available on request.

A Appendix, ports & protocols matrix

Source	Destination	Port / proto	Dir	Encrypted	Auth
Platform	Managed device	161/UDP SNMP	out	v3 (authPriv)	community / USM

Source	Destination	Port / proto	Dir	Encrypted	Auth
Platform	Managed device	22/TCP SSH	out	yes	key / password
Platform	Vendor cloud	443/TCP HTTPS	out	yes	bearer token
Operator / API client	Platform	8080 or 443/TCP	in	TLS	SSO / JWT / API key
Platform	PostgreSQL	5432/TCP JDBC	internal	TLS (recommended)	DB credential
Platform	Batfish sidecar	RPC	internal	private net	n/a (isolated)
Platform	Webhook / SIEM	443/TCP HTTPS	out	yes	HMAC-SHA256
Platform	IdP	443/TCP HTTPS	out	yes	OIDC
Platform	LLM endpoint (optional)	443/TCP HTTPS	out	yes	your API key

B Appendix, read-only collection scope

The collector account needs only read access. SNMP collection issues GET / GETBULK against these MIB families, and the SSH path (opt-in) issues read-only `show`-class commands to capture the configuration text. No OID that would change state, and no configuration command, is ever issued.

Area	MIB / source	Operation
System / inventory	system group, ENTITY-MIB	GET
Interfaces / PoE / stack	IF-MIB, POE-MIB	GETBULK walk
Neighbors	LLDP-MIB	GETBULK walk
VLAN / VRF	Q-BRIDGE-MIB, MPLS-L3VPN-MIB	GETBULK walk
IP / endpoints	IP-MIB, BRIDGE-MIB	GETBULK walk
Routing	BGP4-MIB, OSPF-MIB	GETBULK walk
Multicast / timing / sensors	IGMP-MIB, PTPBASE-MIB, ENTITY-SENSOR-MIB	GETBULK walk
Configuration (opt-in)	SSH	`show running-config` (read)

C Appendix, pre-answered questionnaire map

Where the common assessment instruments (CAIQ/CCM, SIG Lite, VSAQ) ask, this document answers:

Questionnaire domain	Answered in
Identity & Access Management	§12 (SSO, RBAC, grouping, tenant isolation)
Cryptography, Encryption & Key Mgmt	§9, §10, §11
Logging & Monitoring	§14 (tamper-evident audit, SIEM export)
Application / Product Security & SDLC	§16
Threat & Vulnerability Management	§17
Network / Infrastructure Security	§2, §5, §6, §19
Data Security & Privacy Lifecycle	§3, §7, §18
AI / Model Governance	§15
Supply Chain / Sub-processors	Self-hosted; no required sub-processor in the data path (§1, §18)
Governance, Risk & Compliance	§20

D Appendix, security configuration reference

Hardening is driven by environment settings. A representative set of controls follows. The defaults favor working out of the box, while a production setup turns on the fail-closed flags:

Control	Setting	Production
Master encryption key	`CROSSCONNECT_CREDENTIALS_AES_KEY` / `…_KEY_COMMAND` / `…_KEY_FILE`	from KMS or backed-up key file
Token signing secret	`crossconnect.auth.signing-secret`	set, stable
Admin secret	`crossconnect.auth.admin-secret`	set
Require credential on API	`crossconnect.auth.require-credential`	`true`
Require admin secret	`crossconnect.auth.require-admin-secret`	`true`
Fail closed on missing secrets	`CROSSCONNECT_SECURITY_REQUIRE_SECRETS`	`true`
Session lifetime	`CROSSCONNECT_AUTH_TOKEN_TTL_HOURS`	tuned to policy
SSO (OIDC)	`CROSSCONNECT_OIDC_ENABLED` + issuer/client	`true`
Config collection	`crossconnect.discovery.collect-config`	only if needed

CrossConnect by CybrIQ · Security & Architecture Reference · Technical reference · 21 June 2026 · Controls described reflect the shipped operator-preview build; items marked *deployment option* are supported integrations selected at install. · contact_us@cybriq.io