CrossConnect
CONNECT. MONITOR. OPTIMIZE.
BY Cybr IQ

Capability Guide

A plain-English map of what CrossConnect does. It walks every capability in the product, grouped by the area you find it in, and for each one explains what it gives you, how it works, and the data it draws on. Written for technical buyers and evaluators who want an accurate, hype-free survey of the platform.

Audience: technical buyers, evaluators, SRE / network / software engineers

Scope: every navigation area and the capabilities inside it, end to end

Document: capability reference, 21 June 2026

Contact: contact_us@cybriq.io

0 How to read this guide

CrossConnect is a single, trusted record of your network plus the intelligence that reads it. It keeps an authoritative inventory of everything you run, every device, port, address, circuit, and the services they deliver, then works over that record to tell you what is at risk, what changed, and what to do next. It is built for mixed estates, including professional audio-visual (AV) networks, where keeping traffic separated and clocks in sync matters as much as uptime.

This guide follows the product's own menus. Each numbered section is one navigation area. It opens with a short summary and a diagram of how the area fits together, then describes the headline capabilities and lists the rest in a compact table. Every capability shows where to find it in the product as a `route`. Instead of describing a feature with an adjective, each section names how it actually works: the signal, the protocol, the engine, the exact behavior. Capabilities that are an add-on you turn on at install, or an early preview rather than a shipped default, are labelled as such.

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

DEPLOYMENT OPTION integration you enable at install EXPERIMENTAL preview, advisory only

Three ideas used throughout

Read from the gear itself. What CrossConnect shows you comes from reading the real devices over their management protocols, refreshed on a schedule, not from a spreadsheet someone last touched a year ago. Where it is unsure of a value, it labels the confidence rather than presenting a guess as a fact.

Confidence is always shown. When CrossConnect has to make an educated guess (for example, what a given AV device is), it tells you how sure it is and never overstates it: **Confirmed** (the device declared itself, so it is proven), **Inferred** (reasoned from surrounding clues), **Unconfirmed** (a known unknown, paired with the next check to settle it). A value read straight from the live source is Confirmed; a fallback or estimated value is never dressed up as a real measurement.

Read-only and advisory by default. CrossConnect listens to the signals switches already give off (SNMP, LLDP, flow summaries, configs) and the announcements gear already broadcasts. It does **no packet capture and looks inside no traffic**. The AI assistant advises and cites its sources; it never changes your network, and any action that would write a change asks you to confirm first.

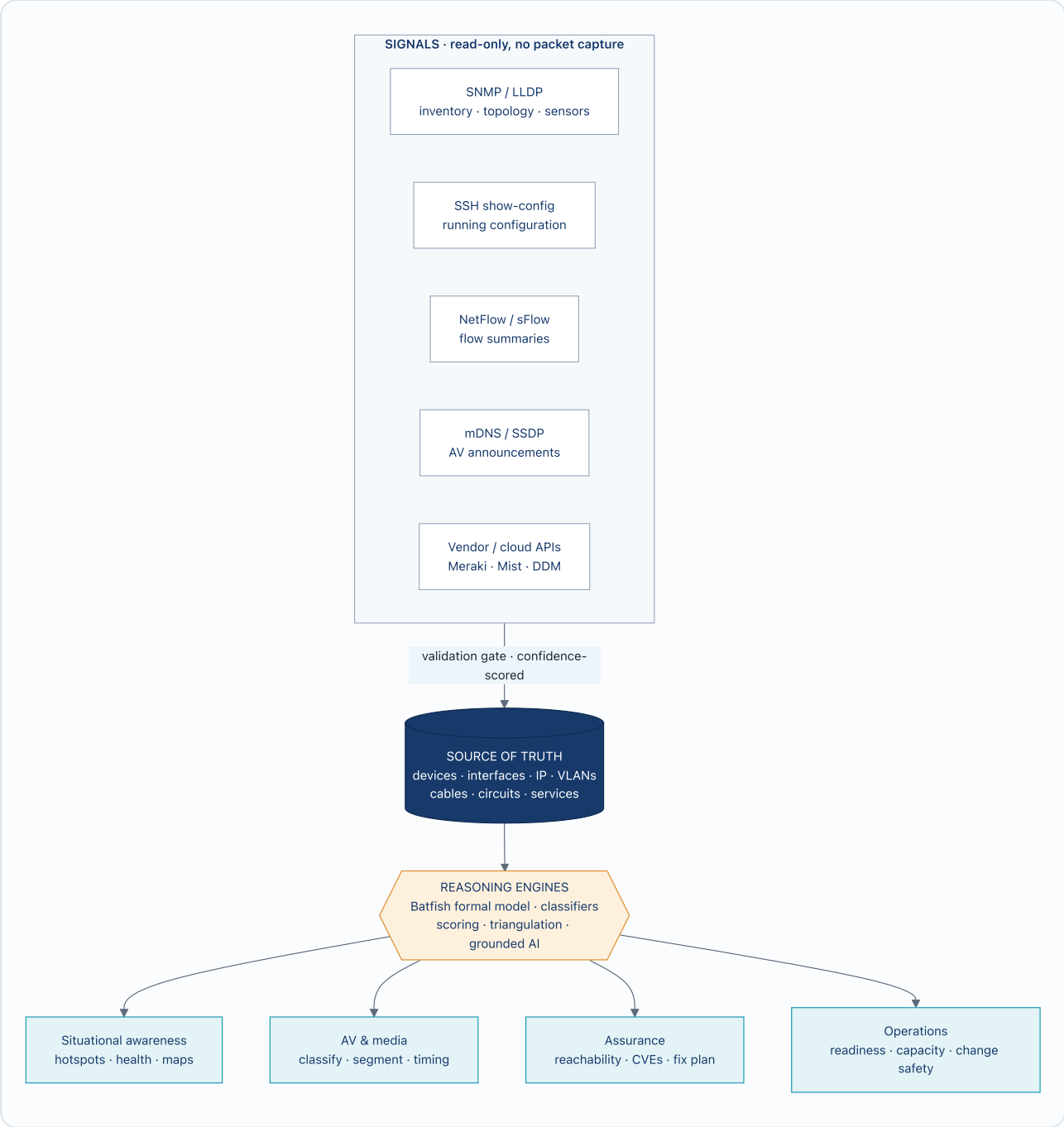

Figure 1. The capability map. Read-only signals pass through the validation gate and are committed into one source of truth. A set of reasoning engines, the Batfish formal model (a vendor-neutral simulation of how the network forwards traffic), the AV classifiers, the scoring and triangulation services, and the grounded AI, then reads that record to answer the questions each area asks. Every capability in this guide sits in one of the four output families.

1 Overview & situational awareness

Where you land and where you look first. This area pulls every risk the platform finds into one ranked list of what to act on now, tells you whether each problem was caused by a recent change, and gives you the live maps and health view you need to get your bearings fast.

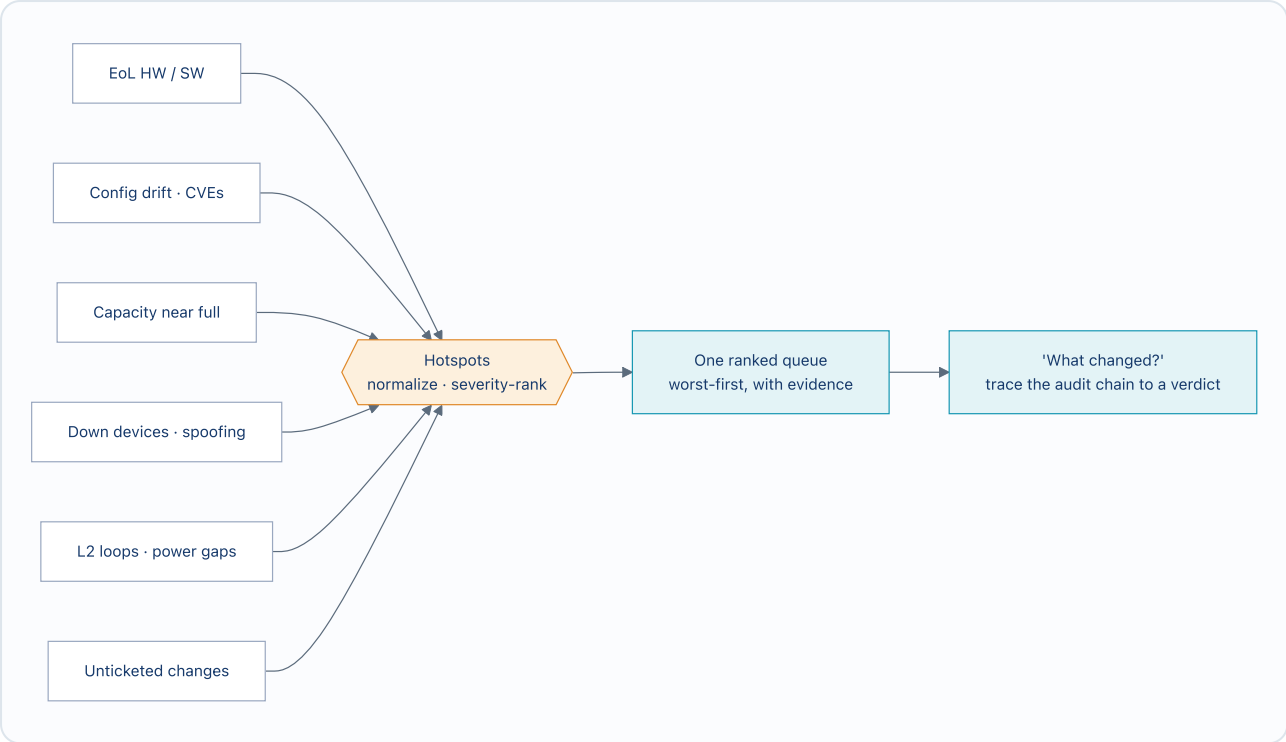

Figure 1.1. One queue, not a dozen dashboards. Every risk the platform finds lands in a single queue ranked by severity, and each row can be traced through the tamper-evident change history to a verdict on whether a change caused it.

Hotspots `/hotspots` GA

One ranked "what should I act on now" queue. It gathers every risk the platform finds, gear and software past end-of-life, configuration that has drifted from intent, known security vulnerabilities (CVEs), capacity nearing full, devices that are down, address spoofing, Layer-2 loops, gaps in power redundancy, and changes made without a ticket, then orders them worst-first. Open any hotspot for a full panel showing the affected object, the evidence, and the recommended next step.

> **What changed, likely cause.** On every hotspot, a trace answers the first question you ask: "did a change cause this?" It walks the tamper-evident history for the affected device or rack and returns a verdict, likely caused by a change, possibly, or no change found (so it is environmental), listing the changes that touched it newest first. It also checks change control: if the suspected change has no authorizing ticket behind it, the verdict flags that. This is read-only correlation over data already held; nothing new is collected.

Topology & maps `/topology` `/map` GA

The Layer-2 wiring map is built automatically from the neighbor announcements switches exchange (LLDP/CDP) and drawn as an interactive diagram. Filter it by vendor, by VLAN, or by a named network service to show only the devices that deliver one outcome and the links between them. The Network map is the geographic companion: your sites pinned on a map, with device counts and drill-down into each location.

More in this area

Capability	Route	What it does
Home	`/`	The landing page: what CrossConnect is and where to start, with quick links into the product.
Health	`/health`	Reachability and environmental telemetry (temperature, fan, PSU) per device, with the unhealthy ones surfaced.
Changes	`/changes`	A recent feed of documented changes across the fleet, so you can see what was touched lately.
Get started	`/get-started`	The first-run guide from a blank install to first value, leading with a zero-risk sample-data path.

2 AV & media

Professional audio-visual gear is built into the model from the start, not bolted on afterward. CrossConnect finds AV equipment on its own and labels each device by what it is (camera, microphone, display, and so on). It then checks that AV is walled off from the rest of the network, that the precise timing every Dante/AES67 audio stream depends on is healthy, and that the shared media and discovery traffic actually flows. All of it runs on signals already collected, with no extra data entry and no packet capture.

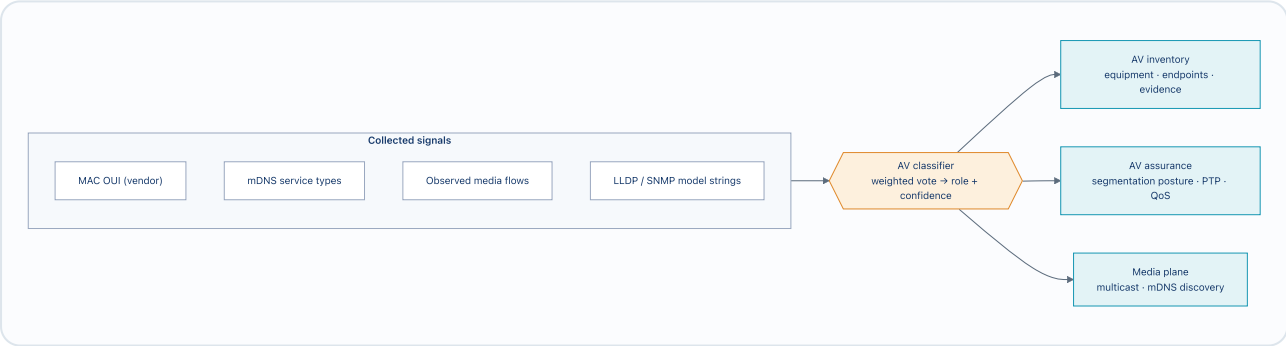

Figure 2.1. One classifier, three lenses. The same collected signals are combined into an AV role with a confidence chip, which then feeds three views: an inventory of the AV fleet, an assurance view of whether that fleet is properly separated and correctly timed, and the media views that confirm the shared traffic and device discovery work. A device must declare itself before it is marked "Confirmed."

AV endpoints & evidence `/av-endpoints` `/av-classify-evidence` GA

Find out what AV gear is on the network and what each piece is, with no data entry from you. CrossConnect labels every AV device by type, codec, display, camera, microphone, DSP, AV-over-IP encoder/decoder, control processor, room PC, speaker, by combining four clues: the maker behind the hardware address (the MAC OUI), the services the device advertises over mDNS, the media traffic it is seen sending and receiving, and the model names it reports over LLDP/SNMP. Those clues produce one role and a Confirmed / Inferred / Unconfirmed chip. AV gear that is not in your inventory is flagged as a shadow AV device. When you want the detail, the evidence view shows one row per clue behind every decision: its kind, its raw value, the role it pointed to, how much weight it carried, the highest confidence it can support, and which collector saw it, so a label is never a black box.

Example: a Shure Dante box that advertises `_dante._tcp` and is sending AES67 audio is labelled av-microphone, Confirmed; an Apple device with no mDNS stays Unconfirmed, shown alongside a concrete next check to settle it.

AV posture & PTP / clock health `/av-posture` `/ptp-health` CONFIGURABLE

AV posture tells you how well your AV gear is walled off from the rest of the network, as a 0–100 score with the reasons itemized. It combines the AV classifier, the zone model, and Batfish reachability into ranked findings, each backed by evidence: shadow AV on a sensitive segment (critical), AV control ports reachable from the user or guest network, cameras or microphones reachable from guest, AV devices straddling two VLANs. The score is honest about what it has checked, reachability findings only count once the formal analysis has actually run. PTP / clock health scores each timing domain and flags the two faults that make audio and video drift: a domain with only one master clock (no backup) and a master clock that has lost lock.

Example: posture 62/100, with one critical (a camera reachable from the guest VLAN); clock-health 70/100, where domain 0 has a single master clock and the domain-1 master is running free, no longer locked (clockClass 248).

QoS & the media plane `/qos` `/multicast` `/mdns-health` CONFIGURABLE

QoS, the rules that give AV traffic priority, is read from each switch's running config into a per-port inventory (policy-maps, priority and bandwidth classes, markings, WRED, the DSCP trust boundary). From that, it finds AV-carrying devices with no priority policy at all, priority classes with no cap that can starve other traffic, and missing trust boundaries. Facts read from config are Confirmed; live queue-drop counts are honestly marked Not Measured until the matching counters (CISCO-CBQOS-MIB) are collected. The multicast suite scores the 239.x media traffic 0–100 across snooping, querier, routing, live delivery, and interoperability, each with vendor-specific fix steps. mDNS / discovery health watches the 224.0.0.251 discovery traffic and finds AV devices split across VLANs with nothing relaying their announcements between them. Every failing item comes with a copy-paste fix in Cisco, Arista, and Juniper syntax; nothing is applied for you.

More in this area

Capability	Route	What it does
AV overview	`/av`	AV as first-class objects: configured Dante/AES67/NDI/Q-SYS media flows with VLAN, bandwidth and latency budget, a redundancy verdict, and the PTP domains.
AV equipment	`/av-equipment`	The documented AV fleet derived from each device's vendor (Crestron, Q-SYS, Extron, Biamp, Shure, Dante): breakdowns by vendor and site, KPI cards, and the AV inventory grid.
AV devices	`/av-devices`	The combined home: documented AV equipment and discovered AV endpoints as two tabs.
AV assurance	`/av-assurance`	The combined home: segmentation posture and PTP clock health as two tabs.

Capability	Route	What it does
Multicast: groups & map	`/multicast-groups`	Every multicast group in plain language, as a list or a flow map, merged from observed traffic, documented AV flows, and IGMP membership; confirmed (seen) vs inferred (documented).
Multicast: issues & fixes	`/multicast-troubleshoot`	The action hub: ranked diagnoses from config/flows/IGMP, each with a vendor-aware runbook, plus per-device posture and unicast-should-be-multicast savings.
AV control-plane sources	`/av-control-sources`	Admin setup for AV controllers fusion reads (Dante Domain Manager, Q-SYS, NDI, Zoom Rooms): read-only endpoint + encrypted token, dormant until configured.

3 Inventory & facilities

The authoritative record of the physical world: every device with its vendor, model, role and software; the modules and spares inside each chassis; and the locations, racks, and power circuits that carry it all. This is the foundation every other area builds on.

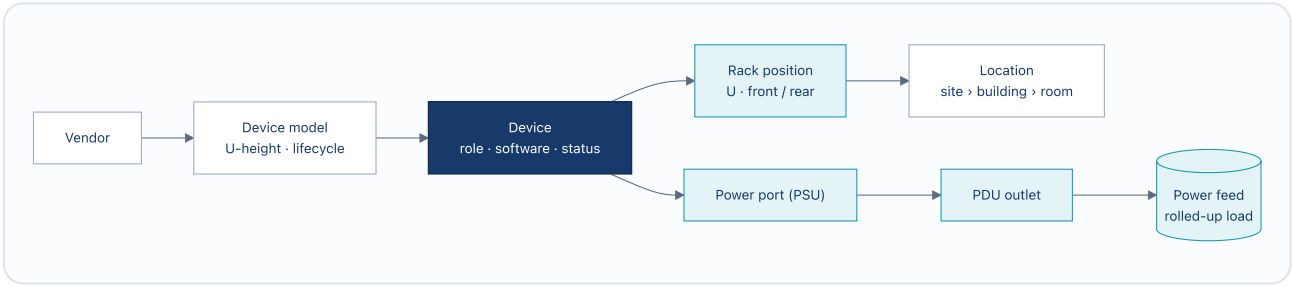

Figure 3.1. Inventory is a chain. Models hang off vendors, devices off models, and every device traces down to a rack position and the power feed behind it, so redundancy and spare headroom are computed from that chain rather than estimated.

Devices & network services `/devices` `/network-services` GA

The device inventory home lists every switch, router, and access point with its vendor, model, location, software version, role, and status. Every column filters and sorts, and this is the object the rest of the platform hangs risk, config, and connectivity off of. Above it sits the service layer: a named outcome the network delivers (for example, "Dante audio fabric" or "Site-B payments"), with the devices, VLANs, and circuits that deliver it tied to it, so health and impact roll up to the business outcome rather than to a single box. Create a service by hand, or accept a suggested one that CrossConnect pieces together from traffic flows, running-service names, and shared VLANs.

Racks & the power chain `/rack` `/power` GA

Rack diagrams track what sits in each rack unit, with drag-to-place, front and rear faces, and half-width mounting so two 1U devices can share one slot side by side. The power chain links a device's power-supply inputs through PDU outlets to a supply circuit (a feed), so you can trace draw from end to end. A feed's total load is the sum of every port that chains up to it, which is exactly what makes the redundant A/B power check in Operations meaningful.

More in this area

Capability	Route	What it does
Device history	`/device-history`	The object-centric timeline for one device from the audit chain: what changed, when, who, and its full location and rack-move trail.
Virtualization	`/virtualization`	Hypervisor clusters and their VMs, each a first-class host with vCPU, memory, disk, a primary IP, and its NICs.
Services	`/services`	Running services on devices and VMs (name, protocol, ports), so "what's open on this host" has an answer.
Vendors	`/vendors`	Hardware vendors, with logos, that device models hang off.
Device models	`/device-models`	Hardware models with U-height, lifecycle, and optional front/rear images.
Roles & platforms	`/roles-platforms`	First-class device roles (core/access/AP) and OS platforms, assignable and filterable.
Module types	`/module-types`	The catalog of pluggable module types (line cards, SFPs) a chassis can hold.
Modules, spares & inventory items	`/modules`	Per-device bays and installed modules, storage media, a spares shelf, and passive items (PSUs, fans, optics) with part and serial numbers.
Device bays	`/device-bays`	Model a chassis as a parent device holding child devices (blades), distinct from module bays.
Locations	`/locations`	The physical hierarchy, sites, buildings, rooms, rows, that everything attaches to.
Power feeds	`/power-feeds`	The electrical supply circuits feeding racks (voltage, amperage, phase, rated watts, safe max), with rolled-up load.
Providers	`/providers`	Carriers and ISPs that deliver circuits, with ASN and portal links.
Custom MAC vendors	`/custom-oui`	Operator-entered OUI-to-vendor mappings that override the built-in table for makers it does not carry.

4 Connections & topology

How everything links together, and what travels along those links. This area covers physical cabling and ports, the endpoints learned off the switches, circuits and paths, wireless and room occupancy, and the analysis that tells you what would lose connectivity if a given device failed.

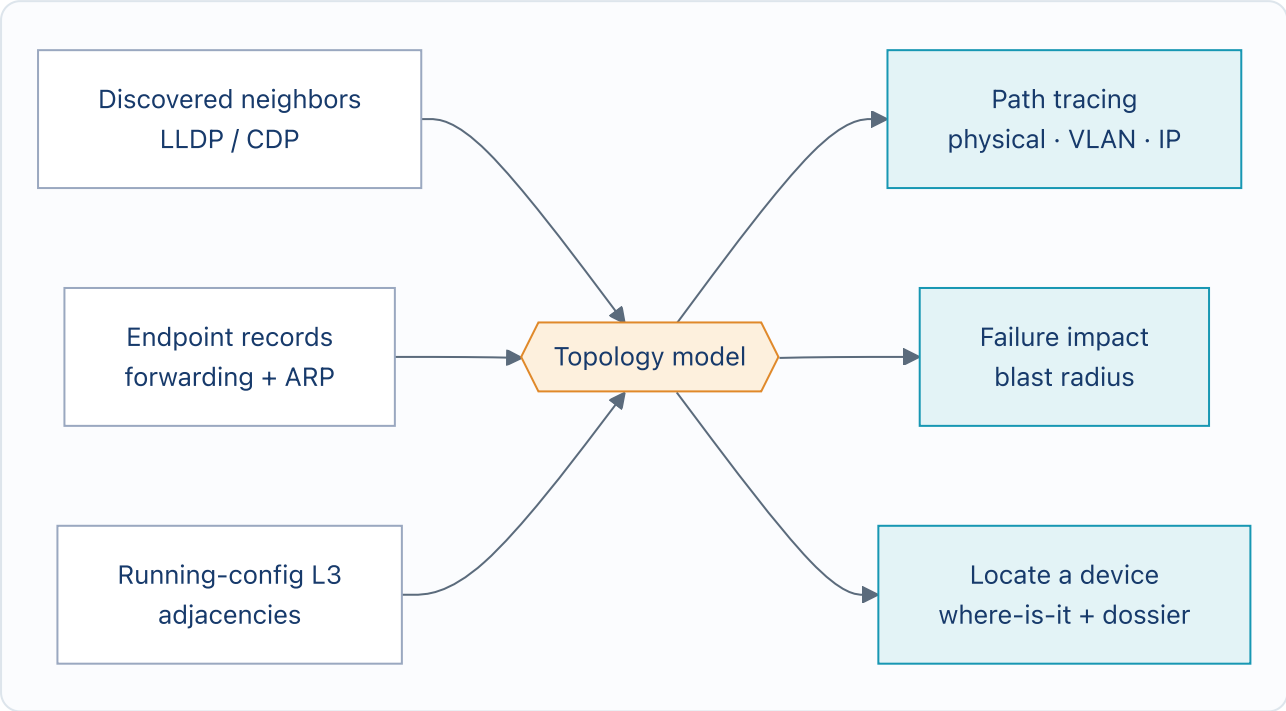

Figure 4.1. Connectivity is discovered, not hand-drawn. Neighbor, endpoint, and config data combine into one topology model that path-tracing, failure-impact, and the locate lookup all run on. Because three independent sources agree on what connects to what, the platform can flag a link the configs create that nobody ever drew.

Locate a device `/locate` GA

Type a MAC or IP address to find exactly which switch and port it sits on, learned from the records switches keep (their forwarding and ARP tables). Then get the full file on that endpoint in one place: what it is doing (observed traffic), whether it has hopped ports, any threat it is named in, and how to reach the switch out-of-band if the network is down. It is a single "where is it?" lookup with no manual entry, plus the context you would otherwise have to gather from four separate screens.

Failure impact & multicast delivery `/failure-impact` `/multicast-groups` CONFIGURABLE

Failure impact answers "what breaks if this fails?" Pick a device, and CrossConnect removes it from a copy of the live network (using the Batfish model) and reports what loses connectivity. An empty result means your redundancy is proven; a list is your real exposure, named alongside the network services that device delivers. Multicast delivery is covered in detail in the AV area; the groups & map view here draws source-to-group-to-listeners from real traffic and IGMP membership, with no packet capture.

More in this area

Capability	Route	What it does
Interfaces	`/interfaces`	The fleet-wide port inventory with channelized subinterfaces and LAG (port-channel) bundling shown under their parent; break out lanes; bond ports.
Cables	`/cables`	Physical links with connector types per end, color, length, and label; discovery proposes most of them.
Patch panels	`/patch-panels`	Passive front/rear ports for modeling structured cabling.
Wireless	`/wireless`	SSIDs (WLANs) and point-to-point wireless links the wired cable model cannot hold.
WiFi coverage	`/wifi-coverage`	A coverage status grid (real client counts, channel plan, co-channel collisions) plus a predictive floor-plan heat map; client counts are real, the heat field is modeled.
Space occupancy	`/occupancy`	Headcount per building/floor/zone derived passively from associated Wi-Fi clients (no cameras, no identity stored), every figure with a confidence band. WHEN INTEGRATED
Occupancy sources	`/occupancy-sources`	Admin setup for wireless platforms feeding occupancy (Juniper Mist, Cisco Catalyst Center): connection details + encrypted token, dormant until set.
Scheduled vs actual	`/schedule-vs-actual`	Compares a room's booked schedule (CSV import) against measured Wi-Fi occupancy hour by hour: no-shows, informal use, over/under-utilization.
Console access	`/console`	Out-of-band console ports and console-server ports, connected so you can trace how to reach a device when the network is down.
Endpoint history	`/endpoint-history`	Every MAC the fabric has seen: current switch/port, first/last seen, moves, and whether it is still present.
Circuits	`/circuits`	WAN/transport circuits with provider, type, commit rate, and A/Z terminations, including fixed-wireless last miles.
Trace · VLAN trace · IP path	`/trace`	Walk a physical path hop by hop, follow a VLAN across the fabric, or compute the L3 path between two IPs.

Capability	Route	What it does
Spanning-tree loops	`/stp-loops`	Cycles in the discovered L2 topology where STP must block a link, with the switches involved.
Forwarding health	`/forwarding-health`	Forwarding loops, ECMP inconsistencies, and BGP/OSPF sessions not established, read passively from the configs (Batfish).
Config topology	`/config-topology`	The L3 adjacencies Batfish derives from running-configs, a third source of truth next to documented cables and discovered neighbors.
Blast-radius pre-flight	`/blast-radius`	Before servicing a device, see the AV streams it carries, rooms that go quiet, PTP domain it anchors, and L3 reachability lost, as a go/caution/no-go verdict.
Network changes (time-lapse)	`/topology-changes`	One map that rebuilds itself: scrub from a baseline to now and watch added gear materialize and retired gear fade.
Relationships	`/relationships`	A graph of how objects connect: devices, cables, circuits, VLANs.

5 Addressing & routing

Your full address plan, from the blocks a registry issued you down to the individual host, plus the VLANs, trust planes, and routing layered over them. The platform works out utilization, overlap, and the next free address as you go; purpose labels let you read the network by what each part is for; and the routing diagram is read from both live device state and what the configs intend.

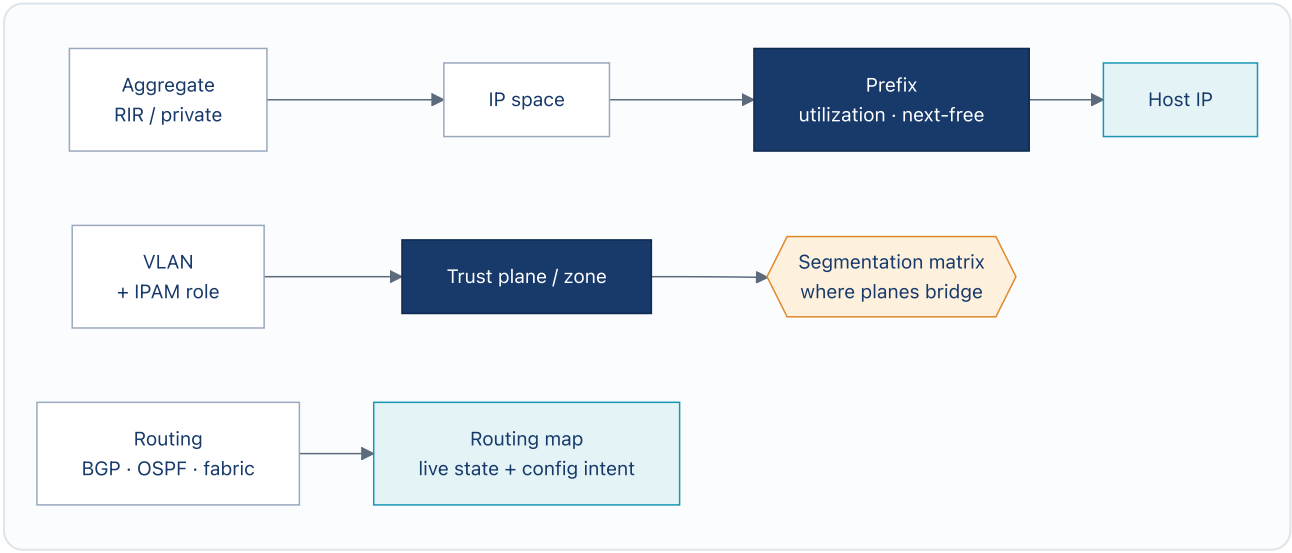

Figure 5.1. From registry block to host, and from VLAN to trust plane. Addresses nest from the largest block down to a single host; each VLAN carries a purpose label that rolls up into trust planes, so separation can be reasoned about; and routing state is read into one diagram that the map and the tables both read from.

Prefixes & network zones `/prefixes` `/zones` GA

Prefixes are address blocks (in CIDR notation) with status, scope, and how full they are, plus overlap detection and a "next free subnet" answer worked out from both documented and discovered hosts. Network zones group your estate into trust planes (audio, video, control, management, data, guest) derived from your VLANs, with a matrix that shows where two planes meet on the same switch. This is the structural view behind the AV posture and segmentation checks: it tells you which planes *should* be kept apart.

Routing: BGP, OSPF, fabric & map `/bgp` `/ospf` `/fabric` `/routing-map` CONFIGURABLE

BGP and OSPF routing sessions are found automatically from two sources, the live device counters (BGP4-MIB / OSPF-MIB) and the Batfish config model, with each row badged to show whether it came from what is documented or what was discovered. An assertions panel calls out sessions that are configured but down, and peerings nobody documented, as clickable findings with fix steps attached. Fabric checks a spine-leaf (Clos) design against the real topology and its BGP numbering (every leaf connected to every spine, no side links inside a tier, a unique number per leaf), and can suggest roles for you to confirm before committing. The routing map pulls all of this into one clickable diagram with a BGP / OSPF / site-level layer toggle, built once by a shared service so the map and the tables never disagree.

More in this area

Capability	Route	What it does
IPAM overview	`/ipam`	IPAM at a glance: one-click next-free-IP with smart defaults, plus a utilization heat list of prefixes filling up.
Aggregates	`/aggregates`	Top-level address blocks your prefixes are carved from, and the RIR or private authority that issued each.
IP spaces	`/ip-spaces`	Top-level address spaces that prefixes and host IPs live in.
IP addresses	`/ip-addresses`	Documented host addresses with status; allocate the next free one.
IP ranges	`/ip-ranges`	Start..end host bands (DHCP pools, reservations) with utilization and a read-only next-free-IP.
VLANs & VLAN groups	`/vlans`	802.1Q VLANs with VID, status, and the carrying devices; groups let a VID repeat across scopes.
IPAM roles	`/ipam-roles`	Purpose labels (audio, video, control, voice, guest, management) shared by prefixes and VLANs.
MAC addresses	`/mac-addresses`	Every documented/discovered MAC resolved to its maker (OUI), grouped for spotting unmanaged gear.
VRFs	`/vrfs`	Virtual routing/forwarding instances with route distinguishers.
FHRP groups	`/fhrp`	First-hop redundancy (HSRP/VRRP/GLBP): the virtual gateway IP and the member interfaces backing it.

Capability	Route	What it does
Tunnels	`/tunnels`	Encapsulated overlays (GRE, IPSec, WireGuard, VXLAN, IP-in-IP) with encapsulation, status, id/VNI, and per-end terminations.
L2VPN	`/l2vpn`	Layer-2 VPNs (VPWS, VPLS, VXLAN-EVPN, MPLS-EVPN) that stitch VLANs or interfaces across sites.
DNS	`/dns`	Forward DNS zones and records for the fleet, completing the name half of the address plan.

6 Assurance & compliance

Proof that the network is correct, secure, and within policy, worked out from the configs and signals the platform already holds. This area answers plain questions ("can a guest reach the cameras?"), pinpoints the change that broke a flow, scores compliance frameworks as data, and rolls every risk into one ranked fix plan.

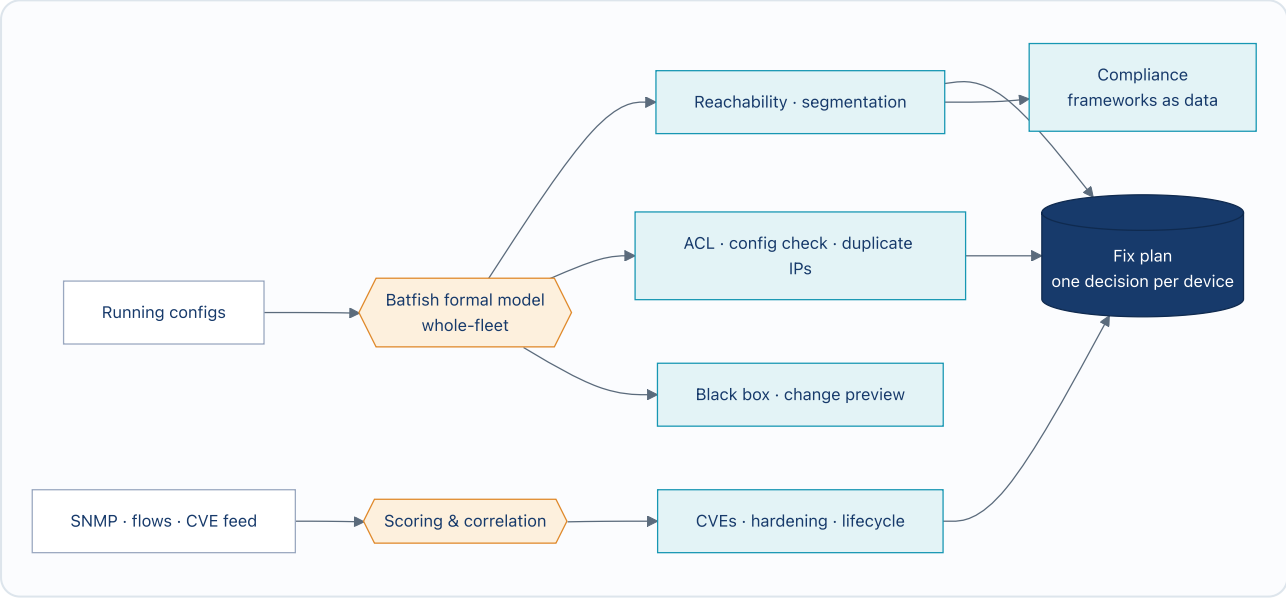

Figure 6.1. One model, many answers, one action list. A single formal model of the whole fleet answers questions about reachability, traffic rules (ACLs), and config correctness; scoring services add vulnerability (CVE), hardening, and lifecycle risk; and the findings come together into one per-device fix plan and reusable compliance evidence.

Reachability, segmentation & the black box `/reach-check` `/segmentation` `/black-box` CONFIGURABLE

The "is AV properly separated?" question, answered two ways. Reachability check tests what you meant to happen: does traffic from A actually reach B through routing and every traffic rule (ACL), and is that what you intended? Segmentation goes further, comparing which trust planes *should* be kept apart against the traffic actually seen between them, so it catches the dangerous case nobody spots today: planes that are isolated on paper but have traffic crossing anyway. The network black box is a flight recorder for your configs: name a flow that used to work and stopped, and CrossConnect searches the saved config history against the formal model to pinpoint the exact change that broke it, returning the before/after difference and the time window as proof. It names the cause; it never changes anything.

Under the hood the black box (`BlackBoxService`) is built from four parts. A **property oracle** states the symptom as a yes/no question (today, reachability: can the source reach the destination on a given port?). A **config timeline** reconstructs the whole fleet's state at any past moment from the saved configuration history. A **bisector** runs a binary search over that timeline (only about log-n checks, the halving trick used to find a word in a dictionary) for the moment the property flipped from good to broken. A **cause report** then names the exact devices and config lines that changed, with the full before and after text and an advisory note on how to reverse it. The property is evaluated by the Batfish formal model, which reasons across Cisco, Arista, and Juniper in one vendor-neutral model, so a failure that crosses vendors is proven, not guessed.

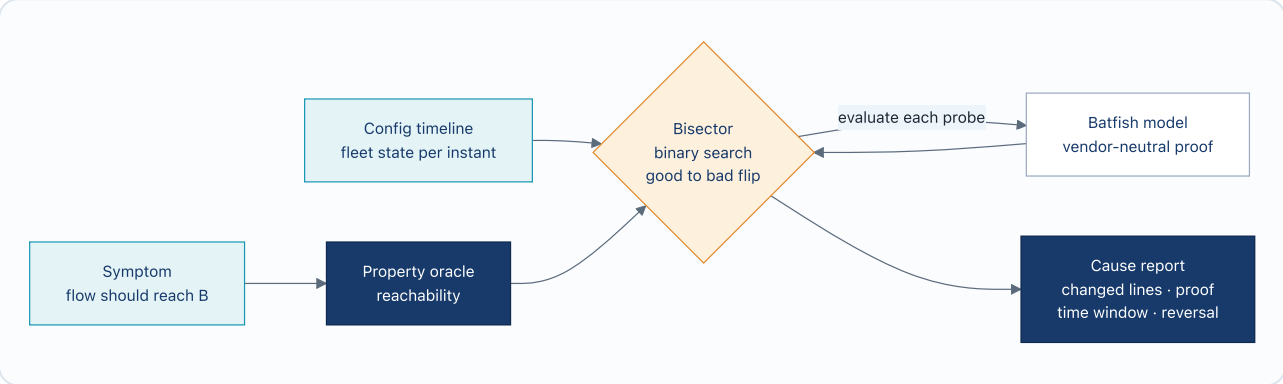

Figure 6.2. The black box, end to end. The symptom becomes a testable property; the bisector replays the saved config timeline through the Batfish model and converges on the interval where it flipped, then names the change with a proof. Strong for Layer 3 reachability (traffic rules, routing, segmentation); Layer 2 mismatches like MTU or VLAN are flagged as suspects, not proven. When nothing in the window broke it, the verdict says so (still healthy, cause predates the window, or inconclusive), which is itself evidence the cause is external.

Example: guest 10.99.0.0/24 to Dante 10.20.0.0/24 UDP 4440: PASS (isolated). The black box shows that path was healthy 48 hours ago and broken by an outbound traffic rule (ACL) added to Vlan20 on cor-bb-01 24 hours ago.

Fix plan & compliance `/fix-plan` `/compliance` CONFIGURABLE

The fix plan is one ranked, per-device to-do list built from several separate risk lists, vulnerability (CVE) exposure, software and hardware lifecycle, and config hardening. Each at-risk device gets a single blended risk score and one recommended action, Replace, Upgrade, or Harden, ordered worst-first, with devices that carry a critical network service pushed up the queue. Compliance scores the controls in each framework (CIS, NIST-CSF, SOC 2, PCI, ISO) as data over evidence the platform already holds, so one reusable check counts toward many frameworks instead of being re-gathered for every audit.

More in this area

Capability	Route	What it does
Security (CVEs)	`/security`	Fleet CVE exposure by severity, mapping advisories to affected software versions.
Threat detection	`/threat-detection`	Spoofing and L2 attacks from FDB/ARP data, IP conflicts, MAC-on-many-ports, and rogues, each with remediation.

Capability	Route	What it does
Config hardening	`/config-hardening`	Per-device security findings (NTP, SSH-only, SNMPv3, AAA), Batfish-derived where available.
Config grade	`/config-grade`	Grades a device config A–F with a score and explains each recommended change.
Config check	`/config-check`	Real config correctness from Batfish's vendor-neutral parse: parse failures, undefined references, and defined-but-unused structures.
ACL check	`/acl-check`	Tests a flow against every ACL in the fleet, reporting permit/deny and the deciding line, with AV presets; also lists dead lines.
Reachability check	`/reach-check`	PASS/FAIL against "should reach" or "should be isolated", with the path.
Change preview	`/change-preview`	Swaps a proposed config into a copy of the fleet and compares reachability, so you see what a change starts or stops before pushing it.
Duplicate IPs	`/duplicate-ips`	IP addresses claimed by more than one device, read from every config, behind intermittent outages.
Controls	`/controls`	Custom compliance controls you define over signal metrics.
Change control	`/change-control`	Every change-controlled event with its authorizing ticket, or a flag when none is attached.
Golden config	`/golden-config`	Running snapshots vs an intended golden config, with drift detection and config diff.
Config contexts & data sources	`/config-contexts`	Scoped JSON that merges onto devices by location/role, synced from a folder or Git.
Vendor analysis	`/vendor-analysis`	Config-level analysis of vendors Batfish does not model (Netgear/Ubiquiti FASTPATH/EdgeOS parsers, Meraki/UniFi cloud API): VLANs, IGMP, ACL/firewall, PoE.
Lifecycle	`/lifecycle`	Hardware and software end-of-sale/-support roll-ups with at-risk devices and recommended targets.
Data quality	`/data-quality`	A score over data-quality checks plus reality-vs-record checks (phantom records, shadow IT, discovery-contradicted cables).

Capability	Route	What it does
Operational maturity	`/maturity`	A CMM-style maturity level from how the platform is actually operated: governance, lifecycle hygiene, data quality, and AV posture where present.

7 Operations

Running the network day to day: whether a site is ready, where capacity is heading, whether a change is safe to make right now, plus the automation, maintenance windows, and reporting that keep the work moving. It all reuses read-only telemetry you already collect, so there is nothing new to install or wire up.

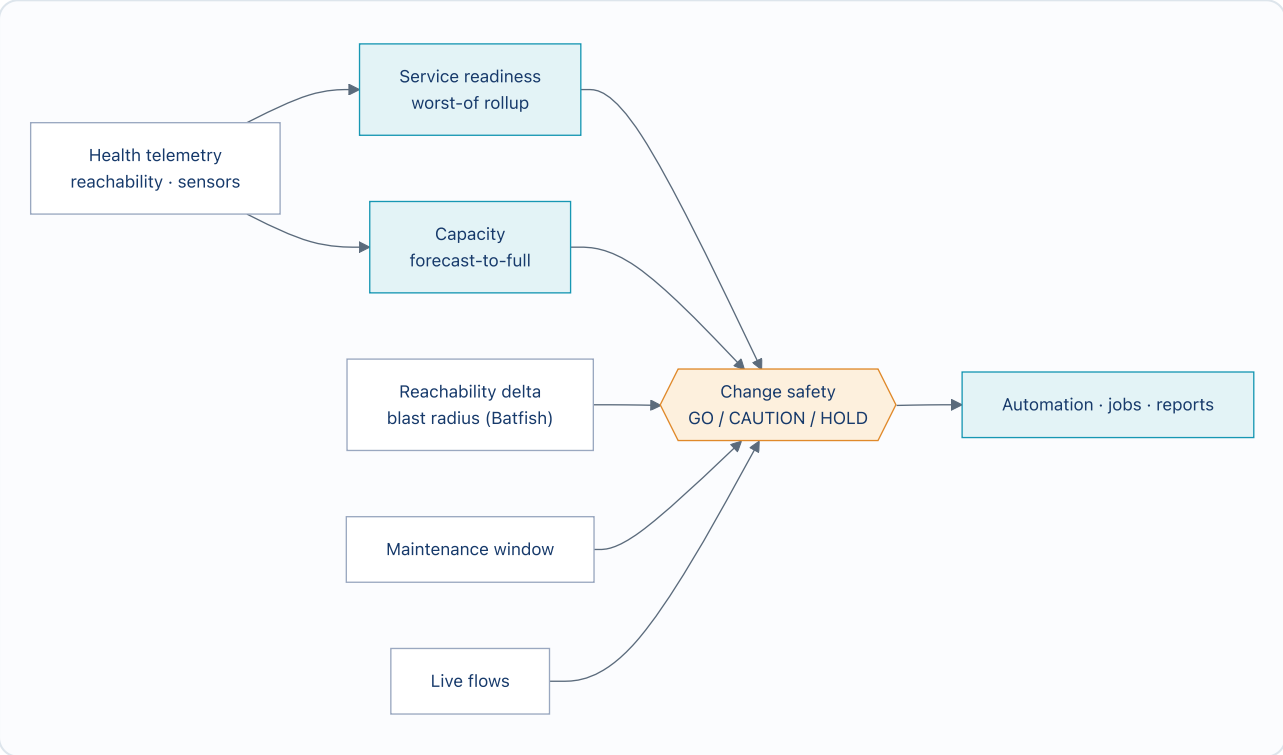

Figure 7.1. One verdict from four independent inputs. The same health telemetry rolls up into both readiness and capacity; change safety then combines what would change about reachability, what breaks if the device fails, what it is carrying right now, and the maintenance window into a single GO / CAUTION / HOLD that no one check could reach on its own.

Service readiness & change safety `/readiness` `/change-safety` GA

Service readiness rolls device health (reachability and environmental sensors) up to the groups that actually deliver a service, by site or by role, so you see whether a whole site is ready rather than just whether one switch is up. A group takes on the state of its worst member, with click-through to the reason. Change safety is the pre-flight cockpit: pick a device, optionally paste in a proposed config, and get one GO / CAUTION / HOLD verdict that combines four things, what would change about reachability, what breaks if the device fails, what the device is carrying right now, and whether you are inside a maintenance window. If the Batfish model is briefly unavailable, the check still runs on what it has.

Capacity planning `/capacity` CONFIGURABLE

How full things are now and when they will run out, across power-over-Ethernet (PoE), rack space, bandwidth, and IP space, with a growth dial and what-if scenarios. A "capacity by service" rollup maps the most-stretched resource on each device onto the network services it supports, worst and most critical first, so you see pressure against a business outcome rather than just against a single device.

More in this area

Capability	Route	What it does
Performance	`/performance`	Interface utilization, throughput, and error rates derived from metric samples.
Traffic flows	`/flows`	Real traffic conversations and top talkers, from a built-in NetFlow/sFlow receiver or a POST API.
Storage	`/storage`	Bootflash, disk, and memory across the fleet (HOST-RESOURCES-MIB) with how full each is.
Rack power	`/rack-power`	Redundant A/B feed power check: neither feed may exceed your threshold so the survivor carries the load.
Automation & event rules	`/automation`	An event-rule engine that fires actions (log, webhook, open a gap) when a signal crosses a threshold.
Jobs	`/jobs`	In-flight and historical background job runs with live output.
Upgrades & executions	`/upgrades`	Plan device upgrades with notification windows and generated IaC (Terraform/Ansible), plus a reviewed, permission-gated execution workflow (request, dry-run, approve, execute).
Maintenance windows	`/maintenance`	Scheduled change/blackout periods over a device, site, service, or tenant, with a live "active now" read.
Reports & scheduled reports	`/reports`	A report catalog with scheduled email delivery; run-now, enable, delete, and a delivery-or-logged state.
Export templates & saved filters	`/export-templates`	Named column presets for CSV/JSON export, and reusable saved filter strings per list view.
Runbooks	`/runbooks`	Operational procedures that apply to many targets (including hotspot kinds), rendered specific to each and exportable to PDF.

Capability	Route	What it does
Journal & documents	`/journal`	A timestamped, attributed operational log you append to any device, and file attachments on objects.
Search & saved queries	`/search`	Full-text search across every record type, plus saved report variants with stored parameters.

8 AI assistant

A chat layer over your whole source of truth. You ask in plain language; it answers from your real records, cites what it used, and keeps documented truth separate from discovered reality. It advises and explains; it never changes the network on its own, and anything that would write a change asks you to confirm first.

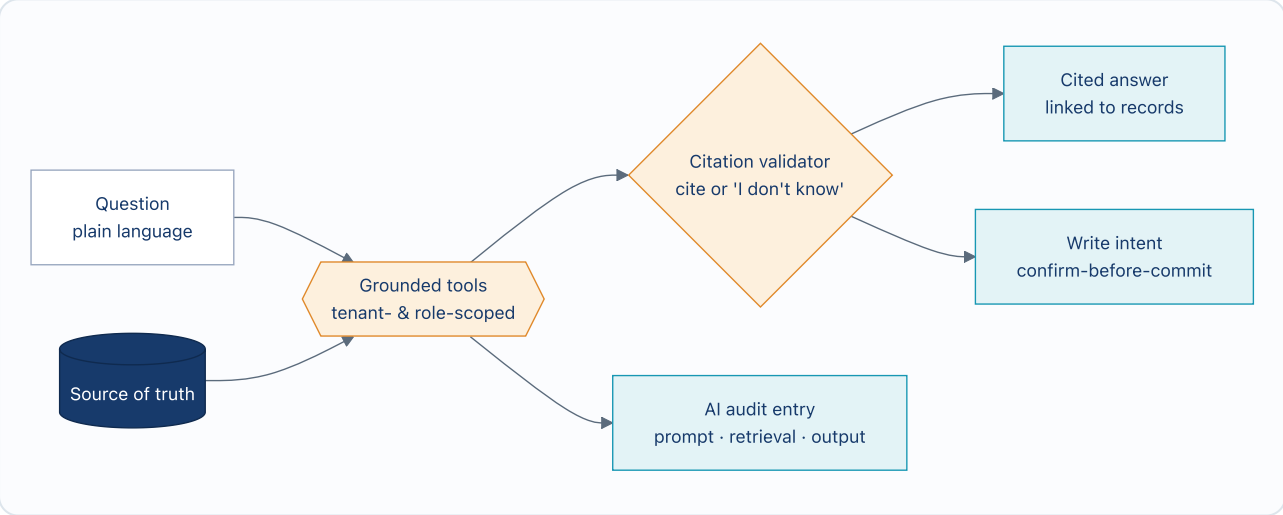

Figure 8.1. Grounded, cited, advisory. Every answer is built from real records and cited. A validator rejects any answer that refers to a record the tools did not actually return, so the model cannot make up devices, IPs, or relationships. Any change is held as a pending request for a human to confirm, and every prompt, lookup, and answer is logged.

Assistant & AI quality `/assistant` `/ai-quality` CONFIGURABLE

Ask about your fleet and get cited, formatted answers drawn from the live source of truth. The assistant respects each user's access rights (it never shows data the user is not allowed to see), reveals the records behind an answer, and logs every prompt, lookup, and answer. AI quality is the improvement loop: a quality score plus a backlog of answers the assistant handled poorly (unsupported by records, held back, or thumbed-down), where each miss points to a tool to add, a feature to document, or a record to fix.

More in this area

Capability	Route	What it does
AI setup	`/ai-setup`	Bring-your-own model: provider, model, and an encrypted key per tenant, with a test-connection. Disable it entirely and the platform returns deterministic non-AI responses.

Capability	Route	What it does
Write intents	`/write-intents`	The queue of AI-proposed changes awaiting your confirmation before anything is applied.
What can I ask?	`/learn/ask-ai`	A searchable gallery of example questions in plain language, what each returns, and why it helps.

9 Administration & governance

The controls that make the platform safe to run and trust: keeping each tenant's data separate, users and roles, a tamper-evident record of every change, the discovery settings that find new gear, encryption-key administration, and connectors that stay off until you turn them on and that pass through a trust review, so nothing from outside quietly becomes truth.

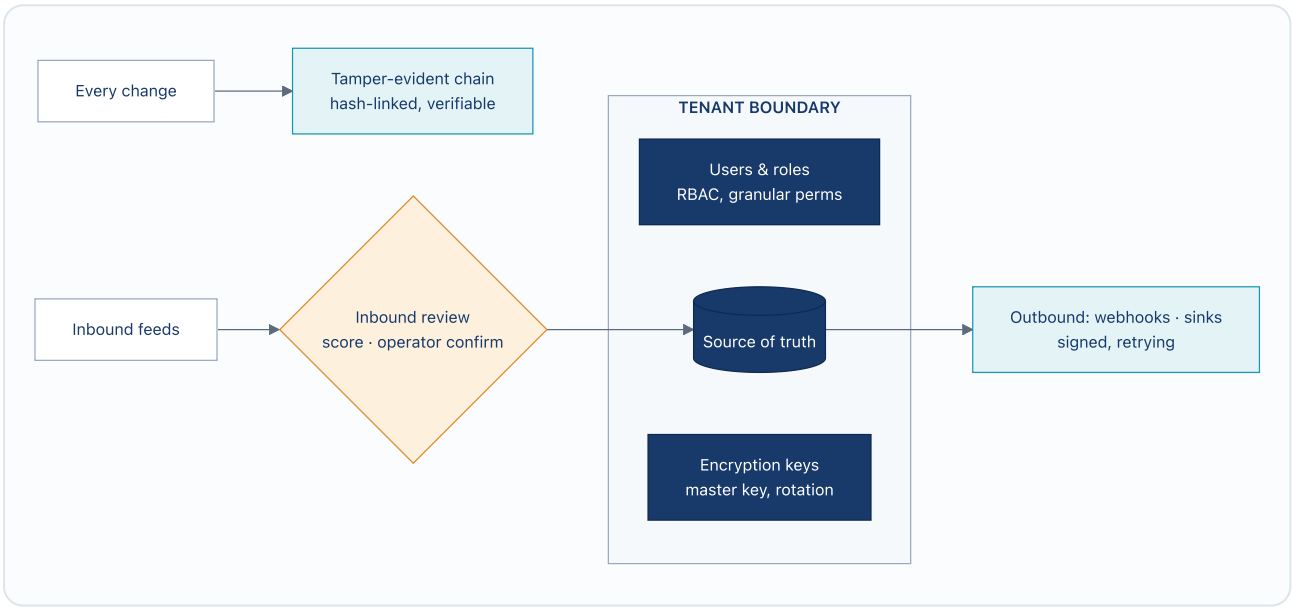

Figure 9.1. Data inside a boundary, change inside a chain. Data sits inside a per-tenant boundary, governed by role-based access (RBAC) and encrypted keys. Every change is linked by a cryptographic hash into a chain you can verify is intact. Inbound feeds stay off until you enable them and are reviewed before they become truth, and outbound feeds are signed.

Audit chain & inbound review `/recent-events` `/logs` `/inbound-review` GA

Every change is recorded into a chain where each entry is cryptographically linked to the last, so you can prove none of it was altered. That guarantee is what backs the "what changed, likely cause" traces and the compliance evidence. Filter by kind and time, verify the chain, and export to CSV. Inbound review is the trust gate for feeds coming in: outside claims arrive as proposals, matched to a known entity and scored for confidence by how well other sources back them up, for you to confirm or dismiss. Nothing inbound quietly becomes truth, two sources that agree score high, while an unknown host scores low and waits for review.

Encryption keys & connectors `/encryption-keys` `/integrations` `/webhooks` CONFIGURABLE

Encryption keys is the administrator view of the master key that protects stored secrets, with a rotation that re-wraps the data key under a fresh master key without having to re-encrypt all your secret data, plus a snapshot taken before rotation that you can roll back to. Integrations lists every

inbound and outbound feed, each off by default and switched on in config, with its status and how to enable it. Webhooks are signed outbound messages on 50+ event types, with automatic retries; outbound sinks are a lighter way to push the activity stream to a security log system (SIEM), Slack, Teams, or PagerDuty.

More in this area

Capability	Route	What it does
Discovery & management ranges	`/discovery-settings`	SNMP/SSH credentials and the management ranges discovery scans; turn scanning on and define the subnets to probe for new gear.
Outbound sinks	`/outbound-sinks`	Point the activity stream at a SIEM HTTP collector and/or a Slack/Teams/PagerDuty webhook, per tenant, with a connectivity test; blank URL = off.
Users & roles	`/users`	Per-tenant users with roles and granular permissions.
API tokens & secrets	`/api-tokens`	Tenant API tokens for the REST surface, and an encrypted vault for device credentials and tokens.
Tenants	`/tenants`	The tenant directory, the isolation boundary for all data.
Custom fields & links	`/custom-fields`	User-defined, typed, validated fields on objects, and templated deep links into external systems (Jira, Grafana).
Tags · contacts · customers	`/tags`	Colored labels, assignable people/teams, and ownership (customers/departments) on any object.
Plugins	`/plugins`	The runtime plugin registry (audit log, webhooks); see which plugins are started.
Trace export	`/trace-export`	One-click runtime on/off for OpenTelemetry trace export; off by default so no collector is needed.
SMTP setup & sitemap	`/smtp-setup`	Per-tenant outbound mail for reports, and a full page map exportable to PDF.
Admin config	`/admin/config`	Tenant-level configuration and toggles.

10 Experimental & Labs

Forward-looking capabilities shipped as clearly labelled previews. Each is read-only and advisory, runs on the same signals and formal model the rest of the platform uses, and is honest about where an educated guess ends. They live under the product's Experimental menu so an evaluator can watch the roadmap working against real data without mistaking a preview for a guaranteed feature.

> **Preview discipline.** Every capability in this section is `EXPERIMENTAL` : advisory only, it never makes a change, never captures packets, and looks the same on live or sample data so you can try it before live signals are wired up. Anything it guesses is labelled as an inference with a confidence level, never passed off as a real measurement.

Gravitational Wobble & Happy Auditor `/wobble` `/happy-auditor`

Gravitational Wobble spots a device CrossConnect does *not* manage by the mark it leaves on the gear it does, the way astronomers find an unseen mass by how it tugs on visible stars. It never probes the unknown device and never captures packets; it reads state already collected (LLDP neighbors, forwarding/ARP tables) and flags a neighbor or endpoint that matches nothing in inventory, pinned to the exact managed port that sees it. Happy Auditor builds a one-click, timestamped, control-by-control evidence pack that an auditor or cyber-insurance underwriter will accept as proof: each control's Satisfied / Gap / Not-applicable status with its evidence, segmentation attestations, the tamper-evident change history, and a verdict on whether the audit chain itself is intact, exportable as self-contained HTML or CSV.

Sense & Red Twin `/sense` `/red-twin`

Sense turns the network itself into a building sensor: room occupancy, AV in use, energy draw, and signs of physical tampering, all inferred from signals the switches already give off (PoE power draw, mDNS AV announcements, IGMP membership, reachability, LLDP), with no cameras, badge readers, or packet capture. Red Twin is a continuous, zero-risk attacker that lives entirely inside the Batfish model and never touches production: starting from a foothold, it searches the model for a path to a high-value target, works out the single traffic rule (ACL) or trunk line that would cut that path (checked in the model to confirm it closes the path and breaks nothing else), and prices each finding against a published cost table (Uptime Institute, ITIC, IBM) as an advisory range, never a quote. It shows the path and writes the fix; it never applies one.

Peek-a-Boo `/peek-a-boo` `/av-displays` `/av-codecs` `/av-dsp`

Peek-a-Boo gathers every network-attached AV camera into one gallery: vendor, model, firmware, location, whether it is online, and whether it is in use right now, drawn from the cameras CrossConnect already classifies. Each card carries risk flags (a vendor barred under NDAA §889, past end-of-support, a known vulnerability) and a "watching" verdict worked out from media bitrate and PoE draw alone, with no video, so a covered lens (BLIND) or a replayed feed (LOOPED) gets caught; a camera missing from the latest scan stays on the board as GONE DARK. Companion views take the same approach to displays and projectors (checked live over PJLink), video-conferencing codecs (a credential-free SIP OPTIONS ping), and DSP/audio gear (Q-SYS cores checked over QRC). Each tries the real thing first: a device it can reach is Confirmed-live; one it cannot falls back to its classification and inventory data.

11 Learn & onboarding

The guided front door to both the product and your own network, plus a library of how-to workflows, end-to-end playbooks, the full API reference, and a map of where every piece of data comes from. Designed so a newcomer can build a working mental model without typing anything.

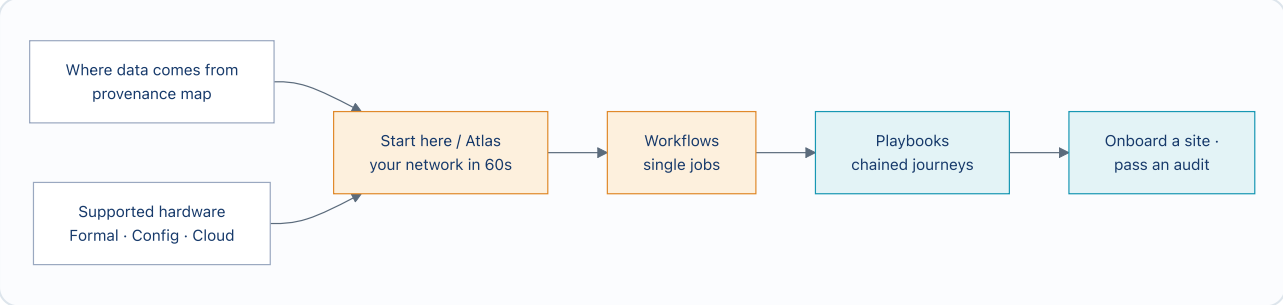

Figure 11.1. Learning scales up. Orient with the Atlas, follow a single workflow, or chain workflows into a full journey like onboarding a site, all read-only against your real estate, with a provenance map and the supported-hardware list grounding every fact.

Start here, the Atlas `/start` GA

A guided front door that reads your real estate and draws its shape, counts, a tier diagram, breakdowns by vendor and role, and a clickable site-and-link map, then previews the operational features read-only against your own network: what is at risk, the AV fleet, what it is carrying, whether it is properly separated, and what is healthy. It also offers a live name/IP/MAC lookup, a built-in grounded assistant, and a clickable first-week path. Everything is read-only, so it teaches both the product and your network with no risk of changing anything.

Where our data comes from `/learn/data-sources` GA

A map of every kind of data CrossConnect holds and where it comes from, SNMP discovery, traffic flows, manual entry, config collection, the Batfish model, external feeds, cross-checking several sources, or pure computation, so you always know whether a fact was discovered, entered, computed, or pieced together. This honesty about origin is what lets the platform keep documented truth separate from discovered reality everywhere else, and it is what backs the AI's citations.

More in this area

Capability	Route	What it does
Learn hub & glossary	`/learn`	The in-product help hub: concept articles and a plain-language glossary so anyone can understand each feature and the terms behind it.
Workflows	`/workflows`	A searchable library of step-by-step workflows for common jobs: where to go, what you do, why, what you need, and the outcome.
Playbooks	`/playbooks`	End-to-end journeys that chain whole workflows toward a goal (onboard a site, stand up service visibility, pass an audit).

Capability	Route	What it does
API & Webhooks	`/learn/api-catalog`	The full REST and webhook reference with copy-paste request, response, and curl examples, plus auth and signing.
Supported hardware	`/supported-hardware`	Every vendor and how each is analyzed: Formal (Batfish model), Config-level (own parser + SNMP), or Cloud (vendor API), rendered live from the canonical catalog.
About	`/learn/about`	System requirements by device count (500 to 100,000), how the architecture scales, and the full software bill of materials.

> **In one sentence.** CrossConnect reads your network over its management protocols, keeps a current source of truth from what it finds, and turns the problems into a ranked, evidence-backed list of what to fix, for the wired estate and the AV estate alike, without touching a single packet.

CrossConnect by CybrIQ · Capability Guide · Technical reference · 21 June 2026 · Capabilities described reflect the shipped operator-preview build; items marked *Experimental* are advisory previews and items marked *deployment option* are integrations selected at install. · contact_us@cybriq.io