VERITY COMMAND

PUBLIC TECHNICAL NOTE

Trust, Authority and Evidence Architecture

How Verity Command establishes who may act, on what basis and within which boundary — and what it does when that basis cannot be established.

DOCUMENT	VERSION	PUBLISHED	EDITION	PDF
VC-PUB-001	1.1	16 September 2026	Public	Download (635 KB)

PUBLIC EDITION

CONTENTS

1 About this note

This note describes the trust, authority and evidence architecture of Verity Command, the sovereign mission intelligence and authorized action platform of Sovereign Inc. It is a public edition: it explains the architecture and its reasoning and omits implementation detail, configuration values and deployment-specific bounds.

It describes how the architecture behaves by definition. It is not a report of operational use, measured performance or independent assessment. Section 12 lists what the note does not claim; the Public Threat Model (VC-PUB-002) sets out what the architecture protects, detects and cannot guarantee, and the Hormuz Reference Scenario (VC-

PUB-003) walks through it with synthetic data.

> Verity Command does not make intelligence inherently true. It makes the basis, authority and consequences of acting on intelligence inspectable and enforceable.

2 Platform and chain

Verity Command governs how intelligence becomes authorized, distributed, executed and evidenced action. Every capability belongs to one stage of a single chain — observe, understand, simulate, recommend, **authorize or refuse**, distribute, enforce, execute, evidence — and the component that recommends never authorizes, while the component that authorizes never executes.

MODULE	ROLE
Reality Fabric	Governed ingestion; provenance, label and lineage fixed at entry
Reality Graph	Temporal and geospatial model of entities, events and decisions, with typed statements
Atlas	Common operational picture; entity and evidence inspection; digest-bound approval
Intelligence	Evidence-linked analysis; governed AI with classification-aware model routing
Simulation	Course-of-action comparison under recorded assumptions; never an observation
Authority Kernel	The single policy decision point; deny precedence; fail closed
Evidence Ledger	Signed, hash-chained records admitted before effect
Tunnel Sovereign	Secure mission distribution bound to authority envelopes
QECNet Trust Fabric	Identity, key lifecycle, policy, revocation and signing on customer-held roots
Verity Edge	Local enforcement of envelope, scope, lease, time and revocation
Verity Forge	Signed releases; configuration and policy as data; sovereign rollout
SDK and Integration Framework	The controlled integration surface; integrations are principals

Deliberate non-goals: autonomous selection or engagement of targets and any weapon-control function; replacement of formally approved force-level command-and-control systems; new cryptographic primitives or proprietary encryption; data brokerage; and mass surveillance. Biometric processing is excluded unless a customer's documented legal authority and a separate product decision exist.

3 Trust boundaries

A deployment is one authority domain: one set of roots of trust and one evidence namespace. Instances interact by federation, never through shared internals. Thirteen boundaries are defined inside the domain; each names the controls a crossing must satisfy and the behavior when they cannot be satisfied.

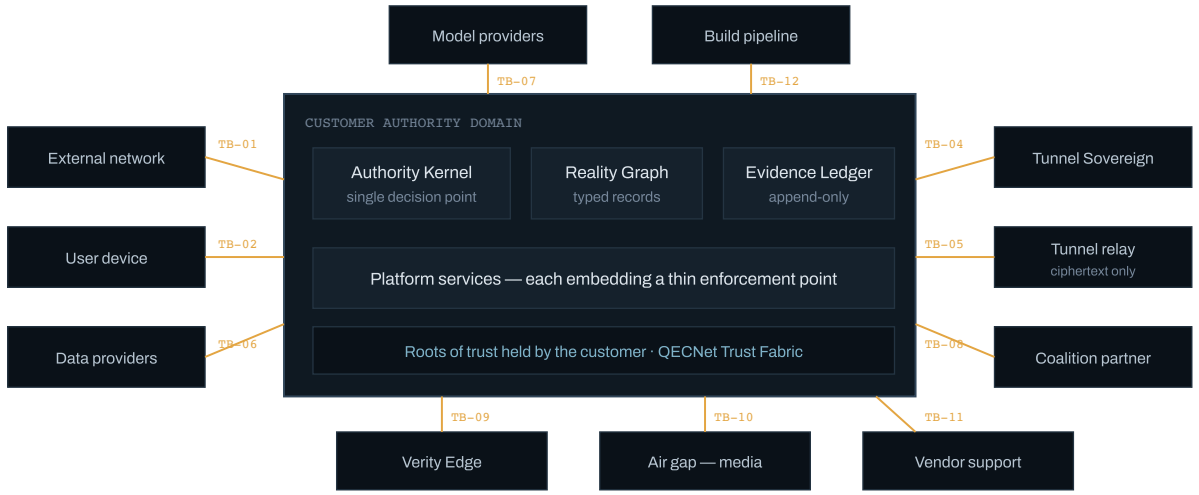

Figure 1 Trust boundaries. Every labeled crossing has its own controls and its own defined behavior on failure. Service-to-service calls and the processing sandbox are internal to the domain.

BOUNDARY	CROSSING CONTROLS	ON FAILURE
TB-01	External network to gateway: strong client authentication, policy decision, rate limits, input validation	Fail closed
TB-02	User device: device-bound session, phishing-resistant authentication, posture checks	Session refused
TB-03	Service to service: workload identity, mutual authentication, per-call decisions	Call denied
TB-04	Platform to Tunnel Sovereign control: signed requests, verification of authority envelopes	Distribution refused
TB-05	Distribution Endpoint to relay: end-to-end encryption to recipients	No content exposure
TB-06	Data provider to Reality Fabric: source authentication, schema validation, scanning, labeling	Quarantine
TB-07	Intelligence to model endpoint: classification-aware routing, egress allowlist	Request denied
TB-08	Graph to coalition partner: release decision, projection, label binding, acceptance	Withhold
TB-09	Authority domain to Verity Edge: signed authority state and leases, conflict preservation	Lease, then fail closed
TB-10	Air gap: signature, provenance and malware checks, import authorization	Import refused
TB-11	Vendor to customer environment: no standing access; approved, time-bound, evidenced	No access
TB-12	Build pipeline to release: signed builds, bill of materials, attestation, customer approval	Deployment blocked

BOUNDARY	CROSSING CONTROLS	ON FAILURE
TB-13	Processing sandbox: restricted profile, no default egress, resource limits	Sandbox terminated

Across every boundary: no component outside a Tunnel Sovereign endpoint holds payload keys; content not releasable to a recipient never crosses a distribution, model, federation or export boundary; the vendor holds no key that can authorize, decrypt or sign inside a sovereign customer domain; every crossing produces a decision record; and untrusted content never becomes instruction.

4 Principals and authority

A principal is any actor that can be authenticated and authorized. People, services, AI agents, devices, sensors, guests and partner domains are distinct principal types with distinct trust anchors in QECNet Trust Fabric. The customer's identity provider remains authoritative for whether a person exists; deactivation there propagates as revocation, and AI agents delegating from that person lose their permissions with them.

ELEMENT	DEFINITION
Grant	Signed conferral of named capabilities within scope, direction and validity window
Scope	Mission, compartment, geography, object set, label ceiling, purpose
Direction	Source to destination; the reverse leg requires its own grant
Delegation	A subset of one's own capabilities, never a superset
Quorum	Number and composition of approvers, set by policy for the action class
Approval	Signed human consent bound to an exact content digest
Epoch	Monotonic version of an authority domain; a grant below the floor is refused

The mission named in an authorization is always the mission of the authorizing grant, never one supplied by the caller; a request naming a different mission is refused. Mission identities use key material that differs per mission, which does not make mission activity unlinkable: connection identity, timing and volume remain observable to infrastructure.

5 The Authority Kernel

There is one policy decision point. No service implements its own authorization; each embeds a thin enforcement point that obtains a decision and applies it. Policy is data — signed, versioned and testable — and the policy engine is a dependency of the Kernel rather than the Kernel itself.

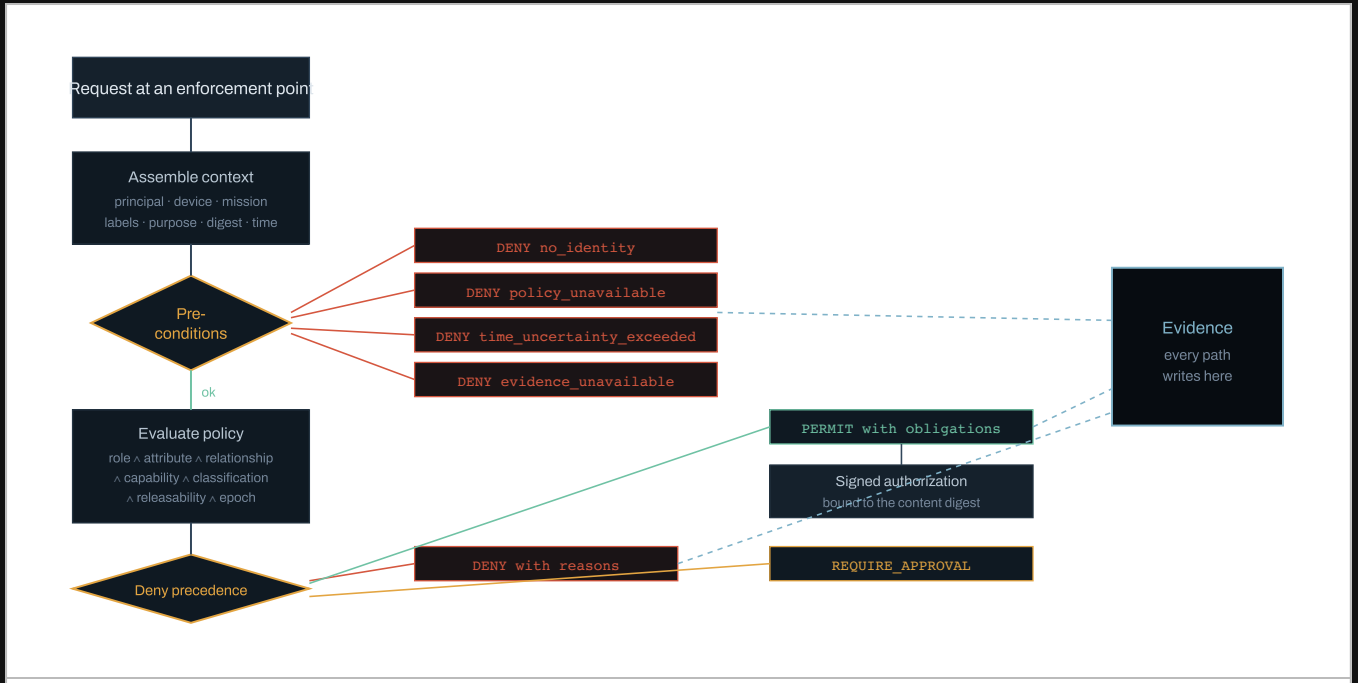

Figure 2 The authority decision path. Four precondition failures produce a denial before any policy is evaluated. Unavailable evidence is one of them: if a decision cannot be recorded, it is not made.

Deny precedence. A decision is permit only if every model permits and no explicit deny matches, so evaluation order cannot change the result. An unrecognized label, category or policy identifier is treated as maximally restrictive.

Fail closed. When identity, policy, evidence, time or cryptographic validation cannot be established, the request is refused and the refusal evidenced. The cost is stated: the Kernel is a single point of denial, and an outage of the decision path stops action rather than permitting it.

Refusal with reasons. A refusal names every unmet condition and the policy version evaluated, so the requester knows what would have to change and a reviewer can reproduce the decision later.

STAGE	WHO	MUST NOT
Recommendation	Intelligence, Simulation, analysts, AI agents	Authorize; execute
Decision	Accountable human commander	Be delegated to an AI agent
Authorization	Authority Kernel with approval and quorum	Execute
Execution	One execution path behind Verity Edge	Act without a verified envelope

A permit produces a signed authority envelope carrying the content digest, mission, principal, scope, direction, validity window and policy version. It is the only object Tunnel Sovereign distributes and the only object Verity Edge accepts, and it cannot be widened or extended by its holder.

6 Evidence before effect

Evidence for a material act is admitted *before* the act's external effect is committed. If the record cannot be admitted, the action does not proceed.

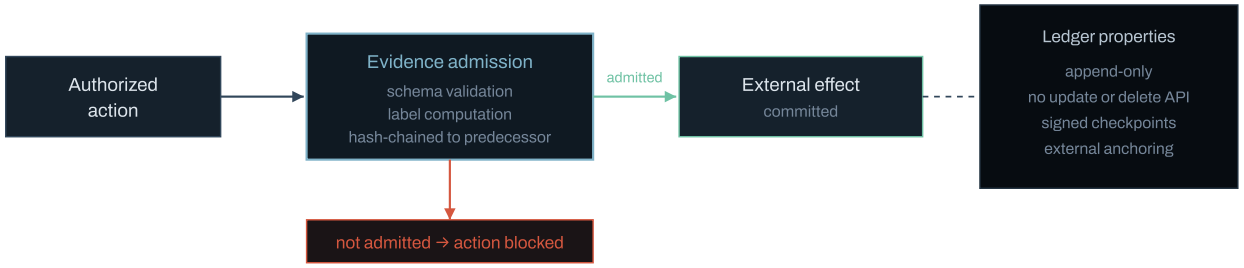

Figure 3 Evidence before effect. The blocked path is not an error to be engineered away; it is the defined behavior when the record of an action cannot be made.

The Evidence Ledger is append-only and tamper-evident. It is not a blockchain: records are hash-chained, periodic checkpoints are signed, and checkpoints can be exported to write-once storage or exchanged with a federation partner so that neither party can rewrite shared history alone. There is no update or delete interface; retention expiry and legal hold are the only removal paths, and both are evidenced. Continuous verification raises a critical incident and freezes writes on any break.

> **What evidence is not.** A complete, verifiable record does not prove that the recorded facts were true, or that a person acted in good faith. It proves what was asserted, by whom, on what basis and when.

7 Time, leases and revocation

Trusted time. Authority is time-bounded, so time sources are a trust boundary. Every record carries its time uncertainty, and unknown uncertainty is recorded as unknown, never as zero. Within bound, leases use the conservative end of the interval; beyond bound, authority decisions fail closed while data collection continues; a detected clock rollback is refused and monotonic counters are relied upon. Specific bounds are deployment configuration.

Leases. A Verity Edge node that loses connectivity continues within a signed lease — capabilities, scope, epoch floors, validity window and policy version — and records its decisions as offline decisions.

LIMIT WHILE DISCONNECTED	REASON
No new grants	Authority originates with the issuer, not the holder
No quorum with absent approvers	An approval that cannot be given cannot be assumed
No lease extension or policy change	A disconnected endpoint cannot grant itself authority

LIMIT WHILE DISCONNECTED	REASON
Fail closed at expiry	Prolonged isolation ends in loss of capability

Reconnection applies authority state first — revocations, epochs, policy — and then re-authorizes queued work item by item. Work whose grant was withdrawn is not released; offline evidence is appended with its local sequence and uncertainty; data conflicts are preserved as contradictions.

Revocation takes effect at an enforcement point when that point receives the signed revocation state, and in every case no later than the expiry of the grant or lease. Connected points honor it promptly. A disconnected point cannot be reached, so its residual authority is bounded by its lease, and the maximum residual window is shown to the commander at the moment of revocation. Revocation is never described as instantaneous everywhere.

8 Provenance in the Reality Graph

The Reality Graph records what kind of statement each belief is. The type is assigned at creation and cannot change.

TYPE	QUESTION IT ANSWERS	WRITTEN BY AI
Observation	What did a sensor, system or person directly detect?	No
Source claim	What does a source state, without having observed it?	No
Machine extraction	What does this content say, as parsed?	Yes
Inference	What follows from these records, by this method?	Yes
Assessment	What does an accountable person judge this to mean?	Drafts only
Prediction	What may happen, under these assumptions?	Yes

Fact is not a type: a statement can be treated as established for a purpose when it meets a corroboration policy, and that treatment is itself a revisable assessment. An inference without inputs is refused at the write path; generated language never promotes a type; confidence is never shown without its method; absence of evidence is shown as absence; contradictions are preserved; and five reports derived from one original are one source, which the lineage shows. Synthetic and exercise data are flagged at creation and can never become observations.

9 Distribution and enforcement

Authorized instructions have to travel and must be checked again where they take effect. The division is strict: **the Authority Kernel authorizes; Tunnel Sovereign delivers; authorized endpoints read; Verity Edge enforces.**

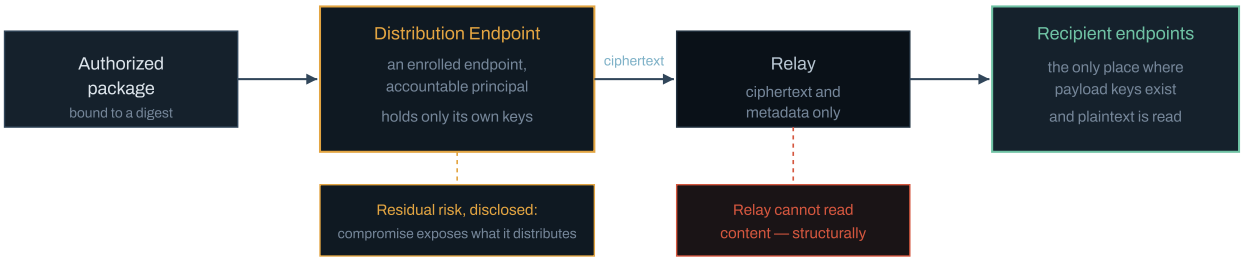

Figure 4 The distribution boundary. The platform cannot encrypt to recipients without being an endpoint, and no central component may hold recipients' keys. The Distribution Endpoint resolves that tension explicitly.

The Distribution Endpoint acts only on a verified envelope, holds plaintext transiently, and runs in a restricted profile without access to the graph, Kernel internals or other evidence. Its residual risk is disclosed: a compromise would expose the packages it is distributing at that time. Mitigations are minimal retention, isolation, separate enrollment, quorum for bulk distribution and volume anomaly detection.

Verity Edge validates the envelope signature and digest, mission scope, its lease, trusted time and local revocation state before releasing an instruction to an operational system, and records acceptance before release and the system's result after it.

Federation is an exchange between two authority domains that each keep their own roots: a release decision by the originating domain and an acceptance by the receiving one. Caveats cannot be stripped, withdrawal is honored, and derived records carry labels at least as restrictive as their most restrictive input unless a downgrade decision is recorded. QECNet Trust Fabric enrolls every endpoint and signs every envelope and record; Verity Forge delivers releases, configuration and policy under the same approval and evidence rules.

10 Failure behavior

CONDITION	BEHAVIOR	WHAT REMAINS
Identity cannot be established	Deny; evidence the denial	Nothing that needs a principal
Policy unavailable	Deny	No extension of existing reads
Evidence cannot be admitted	Deny; action does not proceed	Collection, marked with its state
Time uncertainty beyond bound	Authority fails closed	Collection with recorded uncertainty
Distribution unavailable	Refuse; hold as pending authority	Decision-making; nothing sent silently later
Edge connectivity lost	Enforce within lease; fail closed at expiry	Lease-scoped capabilities, then nothing

CONDITION	BEHAVIOR	WHAT REMAINS
Model route denied or unavailable	Deny rather than reroute	Non-AI analysis
Evidence chain verification fails	Critical incident; freeze writes	Read access to verified history

11 Security invariants

Security properties are written as invariants that hold in every posture, including during failure, and are release preconditions. A representative selection:

- No request is processed without an authenticated principal verified at the enforcement point.
- Every access and action results from an Authority Kernel decision, with deny precedence.
- An AI agent's effective permissions never exceed the intersection of its grant, its delegator's current authority and the mission; no AI agent performs, approves or triggers a high-consequence action.
- Approvals and authorizations are valid only for the exact content digest they bound.
- Leases and validity windows can be renewed only by the issuing authority.
- External and retrieved content is treated as data, never as instruction.

DELIBERATELY NOT AN INVARIANT	WHY
Revocation reaches every endpoint at once	False for disconnected endpoints; the lease is the bound
The relay learns nothing	Connection metadata, timing and volume are observable
A captured unlocked device is protected	It is not; mitigations reduce scope only
AI cannot produce a wrong answer	Governance constrains use, not correctness

12 What this note does not claim

- No certification, accreditation, authorization to operate or approval for classified information.
- No compliance with, or assessment against, any control catalogue, standard, or national or alliance interoperability requirement.
- No independent assessment, audit, penetration test or cryptographic review.
- No measured performance, scale, availability or AI accuracy figure.
- No customer, deployment, endorsement or past performance.
- No comparative superiority over any other system.

> An architecture note that cannot state its limits is not an architecture note. It is an advertisement.

13 Related documents and revisions

VC-PUB-002 · Public Threat Model and Security Boundaries

VC-PUB-003 · Hormuz Reference Scenario — When the Correct Decision Is Refusal

VERSION	DATE	CHANGE
1.0	2026-09-16	First public edition.
1.1	2026-09-16	Aligned to the canonical platform architecture of twelve modules; condensed; related documents linked.

Document VC-PUB-001 · Version 1.1 · Published 16 September 2026 · PUBLIC EDITION