Identity Scope and Continuous Authorization in Brokered Third-Party Access

A design boundary for backend identity; a revocable decision for continued access

Reference note • Version 5.1 • October 2026 • Michael J Martin • outlan.net

Status: Reference architecture and proposed requirements, not a deployment attestation

Abstract

Backend identity non-propagation and continuous access evaluation address different architectural questions. The first defines where the original subject's identity is consumed; the second determines whether an already-authorized session may continue as identity, device, business, and resource conditions change. This note preserves the brokered boundary for unmodified SSH, SFTP, and HTTPS backends while extending the control model from connection-time validation to continuously reevaluated authorization. Products such as SGNL are considered as context, decision, and response-orchestration components, not as substitutes for endpoint telemetry or protocol enforcement. Behavioral modeling of the mediated traffic is treated as a distinct, probabilistic signal class whose attribution depends on the broker's session correlation, and known behavior patterns are governed as versioned, tested signatures. The proposed architecture keeps authenticated subject context in the control plane, binds it to tenant and session state, and requires the broker to implement and verify revocation. The claim is intentionally scoped: a legacy backend can remain identity-opaque, but the access path must not remain policy-static.

Audience: Security architecture, IAM, NAC/endpoint security, proxy engineering, network security, operations, and assurance reviewers.

Evidence boundary: This note describes a reference design and the controls it requires. It is not based on configuration exports, live integration tests, or acceptance results from any specific deployment. Capabilities described here are proposed requirements unless explicitly identified as documented product behavior.

ARCHITECTURAL POSITION

The absence of original-user identity at a backend application does not, by itself, establish a Zero Trust defect. NIST SP 800-207 describes policy enforcement through gateways and portals outside protected resources, while also describing dynamic policy, asset posture monitoring, and continual reevaluation during ongoing communication. A defensible, brokered design must address both the placement of enforcement and its behavior over time [1].

The revised position is therefore narrower and stronger than either blanket claim that “identity must reach the application” or that “nothing is missing.” Original-subject identity may remain within the mediation and control layers, provided the required access decisions can be made and enforced there. Whether the resulting assurance is sufficient remains a use-case decision supported by evidence, not a consequence of naming the pattern Zero Trust.

Table 1. Architectural position by dimension.

Dimension	Architectural question	Proposed position
Identity scope	Where is the original subject authenticated and authorized?	At the broker/control plane; no mandatory original-user assertion to the legacy backend.
Authorization lifetime	How long does an allow decision remain usable?	Only while current policy, required signals, and bounded decision validity permit it.
Enforcement granularity	What can the intermediary actually control?	Session, connection, channel, or request as implemented; no assumed application business semantics.

COMMON CLAIMS AND CORRECTIONS

Arguments for brokered third-party access usually make the right central point: a broker is not obliged to become an identity-native backend. They also tend to overstate it, with language such as “nothing is missing” and “cross-tenant replay is structurally impossible.” This note preserves the boundary argument but removes unsupported protocol generalizations and replaces absolute assurances with explicit control requirements.

Table 2. Common overstatements and the corrected position.

Topic	Common overstatement	Corrected position
Protocol baseline	SSH authentication is reduced to a username, and HTTPS claims are broadened beyond what TLS provides.	Distinguish account identifier, authenticator, lifecycle governance, and authorization.
Tenant identity	SNAT is correctly rejected as identity, but VIP/DNS routing is then elevated into identity proof.	Credentials prove the subject; VIP/DNS select context; policy binds them; SNAT attributes downstream traffic.
Continuous verification	Connection-time OTP and mTLS are described as dynamic re-verification, without a session reassessment mechanism.	Add a signal-to-decision-to-enforcement loop with measurable bounds.
Assurance claims	The boundary is treated as proof of adequacy, without applicable requirements or test evidence.	Separate architectural legitimacy, implementation status, and risk acceptance.

Technical corrections

- **SSH/SFTP:** SSH supports public-key, password, and host-based authentication. The relevant weakness is a deployment that relies on static or weakly governed credentials without sufficiently current authorization, not a protocol that authenticates only a username [2].
- **TOTP:** TOTP is derived from a shared secret and a time step. Public-key registration, if used in this architecture, is a separate enrollment or binding mechanism to document; a 30-second step does not make an active session continuously authenticated [3].
- **HTTPS and mTLS:** Ordinary server-authenticated TLS need not authenticate the client at the TLS layer; application authentication may still exist. mTLS validates the certificate holder under the configured trust policy, not automatically a named human, current device compliance, or entitlement to every resource [4].
- **Tenant separation:** Require the authenticated principal, certificate or account mapping, listener context, requested resource, and policy tenant to agree. Distinct CA names, DNS answers, or source pools are not substitutes for negative isolation tests.

BASELINE BOUNDARY RETAINED

The baseline below is the brokered design this note builds on, not a verified deployment claim. The proposed continuous-authorization extension supplements these controls rather than replacing them.

- **Governed onboarding:** Maintain tenant-scoped identity registration, entitlements, acknowledgments, and lifecycle ownership. An LDAP OU organizes identities; explicit account-to-tenant and resource authorization must enforce the separation.
- **SSH and SFTP mediation:** Authenticate the initiating subject at the broker, evaluate the permitted destination and service, obtain a scoped backend credential from a secrets vault, and establish a distinct backend session. Backend credentials must not be disclosed to the initiating client.
- **HTTPS mediation:** Terminate and inspect TLS at the mediation layer and independently establish the backend connection. Enforce the required mTLS relationships on the enterprise-to-broker and broker-to-enterprise legs; specify and validate authentication for every additional partner/backend leg rather than assuming uniform support.
- **Bidirectional applicability:** Apply equivalent authorization, credential custody, inspection, and revocation requirements to partner-initiated enterprise access and enterprise-initiated partner access. Treat each direction as a separate policy and test case.
- **No bypass:** Prohibit uncontrolled TCP/TLS/SSH passthrough and direct alternative routes to protected services. Verify backend reachability restrictions, listener isolation, management-plane separation, and destination pinning.
- **Downstream segmentation:** Retain tenant-attributed SNAT pools and firewall policy as defense in depth. They constrain the mediated path; they do not prove the original user’s identity or normally identify a single session by themselves.

Identity does not disappear when it is withheld from the backend. The proposed broker must retain sufficient original-subject, tenant, device/workload, resource, and session context for authorization,

revocation, and audit correlation. The backend may authenticate only the broker-selected service account; the control plane must still know on whose behalf that account is being used.

CONTINUOUS POSTURE AND AUTHORIZATION ARCHITECTURE

Where a contextual authorization service fits

This note uses SGNL, acquired by CrowdStrike in 2026, as a documented example. Its Access Evaluations API accepts principal, action, and asset queries and returns Allow or Deny decisions based on configured policy and available business context. Its CAEP Hub documents triggers and actions, including standards-based events, vendor-specific actions, and generic webhooks. These capabilities provide building blocks for the proposed decision and response path; they are not evidence that any particular broker already implements that path [5, 6].

The model should not describe SGNL merely as an endpoint posture scanner. For this design, posture observations originate with the relevant authoritative systems; SGNL or an equivalent service can consume context, evaluate policy, and orchestrate responses. Its documentation distinguishes graph triggers processed after ingestion from a push-based SSF/CAEP event integration, making the ingestion method part of the latency design [7, 8].

Table 3. Mapping of logical responsibilities to candidate components and integration obligations.

Logical responsibility	Candidate component	Integration obligation
Observe and attest state	NAC/asset visibility; EDR/MDM; IdP/directory; business systems	Define authoritative attributes, freshness, device ownership, and identity correlation. Categories are examples, not certified connectors.
Normalize context and decide	SGNL or equivalent contextual authorization service	Map trusted subjects and resources into policy. Return scoped decisions using sufficiently current source data.
Administer response	CAEP Hub and/or a broker integration controller	Translate policy changes into authenticated, tenant-scoped broker actions and invalidate affected cached allows.
Enforce the access path	Protocol-terminating broker acting as PEP	Block new access, reevaluate existing sessions, terminate both session legs when required, and report completion.
Observe session behavior	Flow analytics (NDR) at the ingress and egress collection points; structured broker events; decrypting proxy	Join flows to broker sessions on both legs, emit scored findings with provenance, and never substitute modeling for declared-scope enforcement.
Provide supporting controls	Secrets vault; enterprise firewall/segmentation; SIEM	Scope credentials, constrain reachability, correlate evidence, and support response without pretending these alone terminate application activity.

This is a functional mapping, not a claim that one vendor component exactly implements every NIST role. NIST separates the policy engine’s decision, the policy administrator’s management of the communication path, and the PEP’s enabling, monitoring, and termination of access. The implementation must assign each responsibility explicitly [1].

Figures 1a and 1b show the signal-to-broker control loop and the mediated data path. They appear as landscape panels before the appendix and separate policy decisions, session enforcement, credential custody, and the backend identity boundary.

BINDING SIGNALS TO THE CORRECT ACCESS

A valid signal is useful only if it affects the right principal and the right session. The following is a proposed integration contract, not a representation of native SGNL API fields; broker-side lease, policy-version, and session metadata must be implemented where the product interface does not supply them.

- **Subject and tenant:** Use a stable, issuer-qualified identity plus an authoritative tenant mapping. Preserve principal type: human, service account, device, or workload. Avoid authorization based solely on mutable email names or an asserted tenant header.
- **Device or workload binding:** Bind the authenticated connection to an authoritative endpoint or workload identifier and the relevant credential. Prove how a NAC/EDR/MDM observation refers to this client, not a different device behind the same NAT or another device owned by the same user.
- **Partner-managed assets:** When trustworthy device evidence is unavailable, classify posture as unknown. Use an explicitly approved lower-assurance policy, additional isolation, or deny the protected action; do not convert successful mTLS into an assertion of endpoint health.
- **Resource and action:** Identify the exact protected service and intended operation at the granularity the PEP can actually enforce. If the broker sees only an SSH session, request authorization for that session scope rather than claiming per-command enforcement.
- **Signal provenance and freshness:** Record the authoritative source, source observation time, ingestion time, state/version, and expiry rules. Distinguish compliant, non-compliant, unknown, stale, and conflicting data; absence of an alert is not equivalent to positive compliance.
- **Session and credential correlation:** Maintain an index from subject/tenant/device/resource to frontend and backend session IDs, broker node, policy version, decision lifetime, and scoped backend account or credential reference. Exclude secrets from audit records.
- **Authenticated event handling:** Validate expected issuer, audience, integrity/authentication, and permitted tenant scope. Deduplicate event IDs, handle out-of-order updates, and prevent an older compliant event from restoring access after a newer compromise event.

CAEP standardizes security-event types, including session revocation and device-compliance changes, for cooperating transmitters and receivers. It does not, by itself, define a universal response time or make an arbitrary SSH proxy enforce the event. Subject correlation and enforcement behavior remain part of the integration design [9].

BEHAVIORAL MODELING OF MEDIATED TRAFFIC

The signals in the preceding section describe the state of the subject, device, entitlement, and business context. They do not describe what an authorized session is doing. A partner principal can remain authenticated, entitled, and compliant while its traffic departs from the pattern that justified the grant: a

scheduled SFTP account retrieves an entire directory instead of the expected daily file, an API integration calls endpoints outside its registered scope, or an SSH session requests port forwarding outside its normal window. Behavioral modeling of the mediated traffic supplies that missing class of evidence. NIST SP 800-207 lists network and system activity logs among the inputs a policy engine may consume [1]; this section defines how that evidence is produced for brokered traffic and the limits on how it may influence an access decision.

Behavior is a different class of signal

Posture and entitlement signals are assertions from a system of record: an account is disabled or it is not. Behavioral findings are inferences: a baseline, a model, and an observation window produce a score with a measurable false-positive rate. Table 4 sets out the differences that determine how each class may be used.

Table 4. Comparison of state signals and behavioral findings.

Property	State signal (posture, entitlement, account)	Behavioral finding (mediated traffic)
Origin	Authoritative system of record for the attribute.	Model or detector applied to observed traffic and broker events.
Truth value	Asserted state, subject to freshness and provenance checks.	Scored inference with a confidence level and known error rates.
Subject of the statement	A principal, device, credential, or entitlement.	A session, principal, tenant, or service account over a defined time window.
Latency floor	Source change plus delivery and ingestion.	Observation window plus flow-export timers, collection, and scoring.
Characteristic failure	Stale, missing, conflicting, or unbound data.	False positives, false negatives, baseline drift, and baseline poisoning.
Default use in policy	May directly deny or revoke when fresh and correctly bound.	Graduated response by confidence tier; direct termination only for pre-approved, high-confidence detections.

The unit of observation follows the broker boundary

The brokered design removes original-subject identity from the backend leg. The same boundary determines where behavior can be attributed. The reference design places separate flow collection points on the ingress and egress sides of the broker, with flow records exported from the firewalls. Ingress-side records carry the partner source address and the broker listener; the subject is known only from the broker’s authentication record. Egress-side records carry the tenant SNAT pool, the backend resource, and, through the broker, the scoped service account. Egress telemetry can therefore attribute traffic to a tenant pool but not to an individual subject or session, and ingress telemetry cannot see which backend resource or operation a session reached.

Neither collection point alone answers the operative question: is this subject behaving normally against this resource? The broker’s session-correlation index, which links subject, tenant, device, frontend session, backend session, and credential reference, is the join key that turns the two flow views into one attributable record. The join must use the full address and port mapping on both legs, with synchronized clocks, because SNAT port reuse makes address-only joins incorrect. Without that join,

behavioral analytics is limited to tenant- or pool-level baselines: useful for volumetric anomalies, but unable to separate one compromised account from the rest of its tenant.

Behavioral modeling is therefore a further reason to retain original-subject identity in the control plane. It is not a reason to propagate that identity into the backend.

Observation layers

Table 5 lists the proposed observation layers. Each layer adds features the layer below cannot supply and carries its own limits; none replaces enforcement at the PEP.

Table 5. Proposed observation layers for brokered traffic.

Layer and proposed source	Features available for modeling	Limits to state explicitly
Flow (L3/L4): firewall flow export at the ingress and egress collection points, analyzed in a flow analytics platform	Bytes and packets per direction, duration, peer set, service port, concurrency, inter-arrival periodicity, time of day.	No operation or content. Record timing is bounded by flow-cache timeouts. SNAT aggregation hides individual sessions unless joined to broker records.
TLS and handshake metadata: broker TLS termination and handshake telemetry	Client certificate, SNI, protocol version and cipher, client fingerprint, handshake failure rate.	Characterizes client software and certificate use, not request intent or resource access.
Protocol operations (L7): structured events from broker termination (decrypted view)	HTTP method, path, status, and size; SSH channel types, subsystems, and forwarding requests; SFTP operations (list, read, write, rename, delete), file counts, sizes, and path patterns.	Maps directly to authorization scope, but only if the selected broker emits these events in structured form. Verify per protocol during proof-of-concept testing.
Content: decrypting proxy with file and data inspection	File type, data classification, sensitive-pattern matches, transfer content against declared purpose.	Available only within approved decryption and data-handling scope; Regulated-data handling, retention, and privacy controls apply to the inspection pipeline itself.

Declared intent versus learned baseline

Large third-party estates can include thousands of SFTP relationships. Scheduled file exchanges and API integrations of this kind have a shape that can be stated at onboarding. Governed onboarding should therefore record a declared behavior profile for each tenant integration: direction, protocol, permitted resources and paths, permitted operation set, schedule or window, expected volume and file-count range, expected source endpoints, and principal type.

Two different comparisons follow, and they must not be conflated:

- **Observed versus declared:** A deterministic comparison. An SFTP integration declared write-only that issues reads or deletes, or a session that reaches an undeclared path, is a policy violation. Wherever the PEP can express the declared profile, enforce it as authorization rather than detecting it as an anomaly.
- **Observed versus learned:** A statistical comparison within the declared envelope, for example transfer volume far above the integration's own history, a shift in schedule, a new source endpoint, or an unusual operation mix. This is where behavioral modeling adds value.

The principle is that anything that can be declared should be enforced at the PEP; modeling covers the residual that cannot be written as a rule. A model should never be the only control detecting what an allowlist should have blocked. Interactive human sessions, such as administrative SSH, have weaker baselines; for those, declared scope and step-up authentication carry more weight than learned behavior.

Modeling hazards

- **Cold start:** New integrations have no history. Use the declared profile and peer-group priors from integrations of the same type, and label findings as low confidence until a defined learning period completes.
- **Legitimate change:** Month-end and quarter-end cycles, enrollment periods, partner migrations, and reprocessing after failures change traffic shape. Annotate approved change windows through the onboarding and change process so declared change does not trigger response.
- **Drift and poisoning:** A gradual ramp becomes the new baseline. Bound learned baselines by the declared profile; declared ceilings do not move with learning.
- **Aggregation:** SNAT pools, shared service accounts, and connection reuse merge distinct actors. Model at the joined session level, and report shared credentials as a governance finding rather than tuning around them.
- **Failure storms:** Backend outages produce reconnect and retry bursts that resemble scanning. Correlate with broker and backend health before scoring.
- **Direction:** Model partner-initiated and enterprise-initiated traffic separately, consistent with the bidirectional applicability requirement.

Feeding behavioral findings into the decision loop

A behavioral finding enters the same decision path as any other signal and must meet the same binding and provenance rules, plus fields specific to inference: detector identifier and version, observation window start and end, baseline reference, score and confidence tier, contributing features, and the session-correlation keys the finding applies to. A finding computed on one session or principal binds to that scope; a tenant-pool finding must not revoke unrelated sessions unless the approved response is explicitly tenant-wide.

Response should be graduated by confidence tier and principal type:

- **Observe:** Record and alert without changing access. This is the default for low-confidence findings and during the learning period.
- **Restrict:** Reduce the permitted operation set at the PEP: read-only SFTP, no new channels or forwarding, rate limits, or narrowed paths.
- **Step up:** Require fresh authentication for human principals. Unattended service accounts cannot complete step-up and move to restrict or terminate instead.

- **Terminate:** Revoke and close both session legs using the runtime lifecycle below. Reserve automatic termination for pre-approved, high-confidence detections, such as a declared-profile violation the PEP could not block inline, or for an analyst decision. Pre-approved detections are defined through the signature process in the next section.

Behavioral detection has a latency floor set by the observation window, flow-export timers, collection, and scoring. Flow records are exported when a flow ends or a metering timeout expires [10]. These delays belong in the detection term of the exposure budget. A file transfer can complete inside a single scoring window, so for short transfers behavioral modeling supports detection, response, and forensics rather than prevention; prevention depends on declared-scope enforcement at the PEP. An outage of the behavioral pipeline should raise an alarm but should not by itself deny access, unless a resource's policy makes behavioral monitoring a required condition, in which case it is handled as missing required posture.

BEHAVIORAL ANALYTICS SIGNATURE PROCESS

Learned baselines identify deviation; they do not, by themselves, name what the deviation means. Known-bad and known-required behavior patterns need to be written down as behavioral signatures: named, versioned, testable detections with a defined response. Without a governed process, behavioral detections accumulate untested, are tuned without record, or are wired to responses that differ from what their names and documentation imply. A detection described as terminating sessions but deployed to alert only is the same governance failure as a firewall or IPS rule named as blocking but configured to permit. This section defines the signature process for brokered traffic. It should operate under the same change governance as IPS threat signatures and custom application-identification signatures, not as a separate approval path.

Evidence basis: the decrypted view at the termination point

Behavioral signatures are built primarily on the decrypted view of the session at the broker's termination point. Operation-level behavior — SFTP commands and paths, SSH channel and forwarding requests, HTTP methods, paths, hosts, and status codes, and transferred content — is visible only where the session is terminated and decrypted. For SSH and SFTP this is the broker itself: no passive device can observe operations inside an SSH session it does not terminate. For HTTPS it is the broker's TLS termination and the decrypting proxy.

Firewall flow export, analyzed in a flow analytics platform, observes the encrypted outer flows on each side of the broker. It contributes volume, timing, periodicity, peer, and source-endpoint features, and it corroborates and attributes decrypted-view findings through the session join. It cannot establish what operation a session performed. Two rules follow:

- **Label the basis:** Each signature records whether its logic requires the decrypted view, encrypted-side metadata, or both. A signature that needs operations or content is valid only for integrations whose sessions are terminated and decrypted within approved scope.
- **Cap metadata-only confidence:** A signature based solely on encrypted-side metadata defaults to the Observe or Restrict tier unless it is corroborated by a decrypted-view finding or by declared-profile evidence.

Decrypted-view telemetry carries partner and potentially regulated data, such as health or payment information. The event pipeline that feeds signatures must therefore emit structured operation records and minimal content-derived attributes — classification, file type, pattern match — rather than payload copies, and it falls under the same data-handling, access, and retention controls as the decrypting proxy itself.

The behavioral signature record

Each signature is a controlled record. Table 6 lists the required fields; a signature missing any of them is not eligible for deployment above the Observe tier.

Table 6. Required fields of a behavioral signature record.

Field	Content
Identifier and version	Stable signature ID, semantic version, and change history with rationale for each change.
Scope	Protocol, traffic direction, tenant or integration class, principal type (human, service account, workload), and resources in scope.
Observation layer and features	Required layers (flow, TLS metadata, broker protocol events, content), the exact features consumed, the source of each, and whether each requires the decrypted view at the termination point.
Detection logic	Threshold, sequence, ratio, rate, or deviation from the declared or learned baseline; observation window and evaluation frequency.
Threat mapping	Behavior described and the corresponding MITRE ATT&CK technique where one applies [11].
Confidence and response tier	Confidence tier and the mapped response (Observe, Restrict, Step up, Terminate), with any principal-type exceptions.
Evidence emitted	Fields written to the finding: session-correlation keys, contributing feature values, window, baseline reference, and signature version.
Tests	Positive test cases that must fire and negative test cases that must not, stored with the signature.
Ownership and review	Author, approver, operational owner, review date, and expiry for any scoped exception.

Initial signature classes

Table 7 proposes an initial set of signature classes for SSH, SFTP, and HTTPS mediation. The examples illustrate detection logic and the evidence basis each requires; thresholds are set per integration from declared profiles and shadow-period measurements, not from fixed defaults.

Table 7. Proposed initial behavioral signature classes for brokered traffic.

Class	Example detection logic	Evidence basis	Threat mapping and default tier
Declared-profile violation	Operation, path, schedule, or source endpoint outside the declared profile where the PEP could not block inline, for example a delete on a write-only SFTP integration.	Decrypted view (operations, paths); flow for schedule and source endpoint.	Policy violation; Restrict or Terminate once tested.

Class	Example detection logic	Evidence basis	Threat mapping and default tier
Enumerate then harvest	Recursive directory listing or path enumeration followed within the window by bulk reads exceeding the integration's declared file-count range.	Decrypted view (SFTP list and read operations); flow corroborates volume.	T1083 then T1048.002; Restrict, Terminate on high confidence.
Volume or rate excess	Transfer volume or file count above the declared ceiling, or far above the learned baseline within the declared envelope.	Flow metadata for bytes and timing; decrypted view adds file counts.	T1048.002; Observe or Restrict until baselined.
Credential misuse	Same credential active from an undeclared source endpoint or concurrently from distinct sources; interactive shell use of an SFTP-only service account; authentication-failure bursts.	Broker authentication records and decrypted view (channel type); flow for source endpoints.	T1078, T1110; Step up for humans, Restrict or Terminate for service accounts.
Channel abuse and tunneling	SSH port-forwarding or subsystem requests outside the declared profile; long-lived, low-volume sessions with periodic timing.	Decrypted view (SSH channel and forwarding requests); flow for timing periodicity.	T1572, T1071; Terminate for undeclared forwarding, Observe for timing-only findings.
Probing and cross-tenant requests	Rising 403/404 ratios across paths; requests at one tenant's listener for another tenant's hostnames or resources.	Decrypted view (HTTP status, Host header); TLS SNI visible before decryption.	Reconnaissance (T1595); Observe, escalate on repetition. Rejected cross-tenant attempts also feed tenant-isolation evidence.

Signature lifecycle

Every signature moves through the stages in Table 8. Each stage has an exit criterion, and the record carries the evidence for each transition.

Table 8. Behavioral signature lifecycle stages and exit criteria.

Stage	Activity	Exit criterion
1. Intake	Capture the candidate from threat intelligence, incident lessons, red-team results, proof-of-concept findings, onboarding declarations, or repeated model findings. Open a tracked change record.	Behavior described, scope stated, and owner assigned.
2. Policy-first check	Determine whether the PEP can enforce the behavior as authorization. If it can, route it to policy and close the signature request.	Documented reason the behavior cannot be enforced inline.
3. Authoring	Complete the signature record, including detection logic, threat mapping, confidence and response tier, and evidence fields.	All required fields present; peer review completed.
4. Data validation	Confirm every required feature exists at the needed granularity on both legs, including the flow-to-session join and structured broker events. For decrypted-view features, confirm that approved decryption scope covers the integration.	Feature availability demonstrated on production-representative telemetry.

Stage	Activity	Exit criterion
5. Testing	Replay positive cases per protocol and direction; replay negative cases drawn from legitimate traffic, including period-end peaks and retry storms. Measure detection delay.	All positive tests fire, negative tests stay silent, and delay is within the approved bound.
6. Shadow deployment	Run at Observe for a defined period; analysts disposition every finding by tenant and integration class.	False-positive rate and alert volume within approved limits.
7. Approval	Architecture, SOC, and risk owners approve the confidence and response tier. The author does not approve their own signature.	Recorded approval for the specific version and tier.
8. Promotion and action verification	Deploy through the controlled pipeline, then verify that the deployed action matches the approved response tier in configuration and in a live test.	Configured action, observed action, and approved tier agree.
9. Operation and tuning	Track per-signature findings, dispositions, true-positive rate, detection delay, and verified response outcomes. Every threshold change is versioned, justified, and retested.	No unversioned change; metrics reported on the review cycle.
10. Review and retirement	On each review date, confirm continued relevance. Retire signatures that never fire on valid tests, are superseded by PEP policy, or depend on retired telemetry.	Signature renewed, revised, or retired with a recorded reason.

Controls on the process

- **Signatures as code:** Store signature records, detection logic, and test fixtures in version control and deploy through a pipeline that runs the stored tests. Rollback restores the previous approved version, not an ad hoc edit.
- **Suppression is a signature change:** Tenant- or integration-specific suppressions and exceptions are scoped, owned, and time-bound, and expire automatically. A suppression without an expiry is not permitted.
- **Configured-action audit:** Periodically compare every deployed signature's configured action with its approved response tier. A mismatch is a finding against the signature process, independent of whether the signature has fired.
- **Escalation path:** Learned-model findings that repeatedly prove true are codified as signatures. Signatures whose logic becomes deterministic and enforceable at the PEP migrate to authorization policy. Detection should move toward prevention wherever the enforcement point allows.
- **Metrics that matter:** Report true-positive rate, analyst disposition time, detection delay, and verified response completion per signature. Alert counts alone do not show whether the process is working.

RUNTIME AUTHORIZATION LIFECYCLE

Admission and continued access

- **Admission:** Authenticate the caller and validate tenant/resource binding. Evaluate current policy using required fresh signals, then acquire the least-privileged backend credential and open the mediated path only after an allow decision.
- **Decision lifetime:** Assign a bounded broker-side authorization lifetime appropriate to resource risk. Reevaluate before expiry and before any newly exposed higher-risk operation; never let connection reuse silently extend an expired decision.
- **Change-driven reassessment:** Account disablement, entitlement withdrawal, device non-compliance, a credible compromise signal, closure of an approved work item, resource reclassification, and relevant policy changes must identify and reevaluate affected active sessions.
- **Revocation ordering:** Invalidate cached permission and block new/replacement access before, or atomically with, closing affected existing sessions. Record a revocation generation so a delayed admission response cannot reopen access under an older decision.
- **Execution and verification:** Send a scoped termination or restriction request to the owning broker nodes, retry idempotently, and verify the actual data-path result. An event delivery receipt or successful webhook response is not sufficient evidence that access stopped.
- **Recovery:** After remediation, require a fresh policy evaluation and, where warranted, fresh authentication. Do not automatically reopen a terminated session solely because a later event reports compliant status.

Push events plus bounded fallback

Use event-driven reassessment for rapid response and periodic or per-request reevaluation as a backstop for missed events, lost subscriptions, cache drift, and policy changes. The fallback must obtain sufficiently current authoritative state; polling an unchanged stale cache does not close an event-delivery gap. The design should also reconcile after broker failover and restart rather than treating recovered sessions as implicitly trusted.

SGNL documents that graph triggers are processed after ingestion completes and that underlying synchronization frequency matters for low-latency response. A fast policy API therefore does not prove fast source-to-session revocation. Procurement and proof-of-concept testing must verify the actual ingestion mode, broker hook, and completion semantics for every selected feed and enforcement path [7].

PROTOCOL-SPECIFIC ENFORCEMENT BOUNDARIES

Table 9. Protocol-specific continuous behavior and limits to state explicitly.

Protocol/path	Required continuous behavior	Limit to state explicitly
HTTPS requests	Reevaluate each protected request or use a short, bounded and invalidatable decision cache. Bind application-session and cookie handling to the broker decision.	TLS establishment is not authorization for every later request; an upstream pool must not erase user/tenant attribution.
WebSocket, HTTP/2 streams, or long downloads	Map active streams/connections to the authorization subject and close or restrict them when policy requires.	Rejecting the next HTTP request alone does not stop an already-open stream.
SSH	Maintain frontend/backend session and channel mappings. Deny unapproved forwarding and subsystems. On revocation, stop new channels and close affected existing channels and both session legs.	Do not assume transparent mid-session MFA or per-command controls. If unsupported, terminate and require a newly authorized connection.
SFTP	Stop affected transfers and prevent reconnect/resume under a revoked decision. Define partial-file handling and an application-safe retry/reconciliation process.	Already delivered bytes cannot be recalled; a completed write is not undone by terminating the transport.
Backend work and credentials	Constrain backend privileges; test whether disconnect leaves jobs or child processes running. Use separate backend cancellation controls when required.	A decision change or vault credential rotation is not evidence that an authenticated session or detached backend job has stopped.

These are proposed requirements and test boundaries, not assertions that every proxy exposes the necessary APIs. Where the selected broker cannot enforce the required granularity, choose a stronger intermediary, reduce the permitted operation set, or record an explicit residual-risk exception. Do not relabel a missing enforcement mechanism as an acceptable identity-propagation boundary.

FRESHNESS, FAILURE BEHAVIOR, AND MEASURABLE BOUNDS

Use “continuous” to mean ongoing, change-responsive authorization with documented maximum staleness and enforcement bounds, not literally instantaneous or constant scanning. No production timing values are assumed here. Architecture and risk owners must approve the bounds before enforcement acceptance.

```
Exposure budget = detection + delivery/ingestion + evaluation
                + dispatch/retry + PEP enforcement/verification

Fallback bound  = refresh/reevaluation wait + processing
                + PEP enforcement/verification
```

Treat these as conservative budget models whose terms must be measured on the implemented path. Source polling, queueing, cache validity, clock error, and retry limits belong in the relevant terms; event-path and fallback-path performance should be tested separately. Detection delay may be unbounded for an unobserved condition, so a response SLA must identify its start event and cannot promise immediate detection of all compromise.

Table 10. Recommended defaults for protected access by failure condition.

Condition	Recommended default for protected access
Fresh authoritative deny or compromise signal	Deny new access, invalidate affected allows, and revoke active access within the approved bound.
Missing, stale, conflicting, or unbound required posture	Treat as insufficient evidence. Deny new sensitive access; expire or terminate active authorization according to the approved stale-data policy.
Policy service or source outage	No new unvalidated grants. Any cached-allow grace must be explicitly risk-approved, scoped, non-renewable without fresh evidence, and capped by a hard expiry.
Lost event channel	Alarm and activate current-state reconciliation; enforce the same hard freshness bounds. Do not continue indefinitely on last-known-good state.
Broker action failure or failover	Retry and reconcile session ownership; escalate failed enforcement. Use validated network containment if available and necessary, recognizing possible shared-tenant collateral impact.
Behavioral anomaly finding	Respond according to the approved confidence tier and principal type. Low-confidence findings alert only. A behavioral-pipeline outage alarms but does not deny unless policy makes monitoring a required condition.
Break-glass access	Use a separate approved, time-limited, strongly authenticated and audited emergency process. An outage must not silently become a global fail-open policy.

A posture source outage and a policy-service outage are different failures and need separate alarms. Record source-change, ingest, decision, action-request, and verified-completion timestamps so latency can be attributed and tested; report normal response distributions as well as failures and worst observed delay.

OPERATIONAL OWNERSHIP AND ACCEPTANCE GATES

Adopt the extension in stages: integration inventory, shadow evaluation, new-session enforcement, active-session revocation, and failure-mode validation. Shadow mode may establish policy quality and signal correlation, but it must not be described as enforced continuous authorization. Promotion requires evidence for both traffic directions and each protected protocol.

Table 11. Acceptance gates and evidence required by owner.

Gate / owner	Evidence required before acceptance
Identity binding / IAM + endpoint teams	Map a user on two devices and two tenants correctly. A posture change on one device affects only its bound sessions unless the event explicitly has user-wide scope.
Tenant isolation / PKI + proxy teams	Reject Tenant-A credentials at Tenant-B listeners, wrong-host/resource requests, and unauthenticated tenant assertions. Prove no route, trust-store, credential, or pool crossover.
Source freshness / signal owners	Demonstrate polling and push delays, observation timestamps, expiry behavior, source outage detection, and unknown-device handling.
Admission / policy + proxy teams	Prove deny creates no backend path; demonstrate that policy changes and revocation invalidate cached permits and block reconnect races.
Live revocation / proxy team	Disable a user or change posture while SSH, SFTP, HTTPS, and long-lived streams are active. Verify both legs and all affected channels stop within the approved bound.

Gate / owner	Evidence required before acceptance
Transfer/process safety / application owners	Test partial SFTP writes, retries, long downloads, and detached backend work. Document what stops, what remains committed, and required cleanup or cancellation.
Resilience / platform operations	Lose an event, replay or reorder events, stop the policy service, and fail over a broker node. Prove finite expiry, reconciliation, and no silent fail-open.
Credential scope / vault + application teams	Verify backend account least privilege, secret non-disclosure, rotation behavior, and blast radius if a broker or service credential is compromised.
Audit / SOC + assurance	Correlate original subject through tenant, signal, policy decision, broker sessions, backend account, and confirmed outcome without logging credentials.
Behavioral analytics / NDR + SOC + proxy teams	Prove the flow-to-session join on both legs, including under SNAT port reuse. Replay synthetic deviations: off-schedule transfer, volume excess, undeclared operation, new source endpoint, and forwarding attempt. Report the shadow-period false-positive rate and measured detection delay.
Signature governance / SOC detection engineering	Show each promoted signature's complete record, passing stored tests, shadow-period dispositions, approval by someone other than the author, and agreement between configured action, observed action, and approved tier.
Approval / architecture + risk owners	Approve freshness/revocation bounds, action-level limitations, exceptions, emergency access, and runbooks. Record measured results rather than product capability claims.

Outstanding design decisions include the selected broker's session-control interfaces, supported SGNL integration or adapter path, authoritative posture sources, trusted partner-device binding, resource/action granularity, approved failure/timing policies, the flow-to-session join method, declared behavior profiles, behavioral response tiers, and the signature repository, test pipeline, and approval workflow. These are implementation dependencies, not evidence that original-user identity must be sent to the backend.

TERMINOLOGY AND REVIEW-READY STATEMENT

Table 12. Terminology to avoid and recommended replacements.

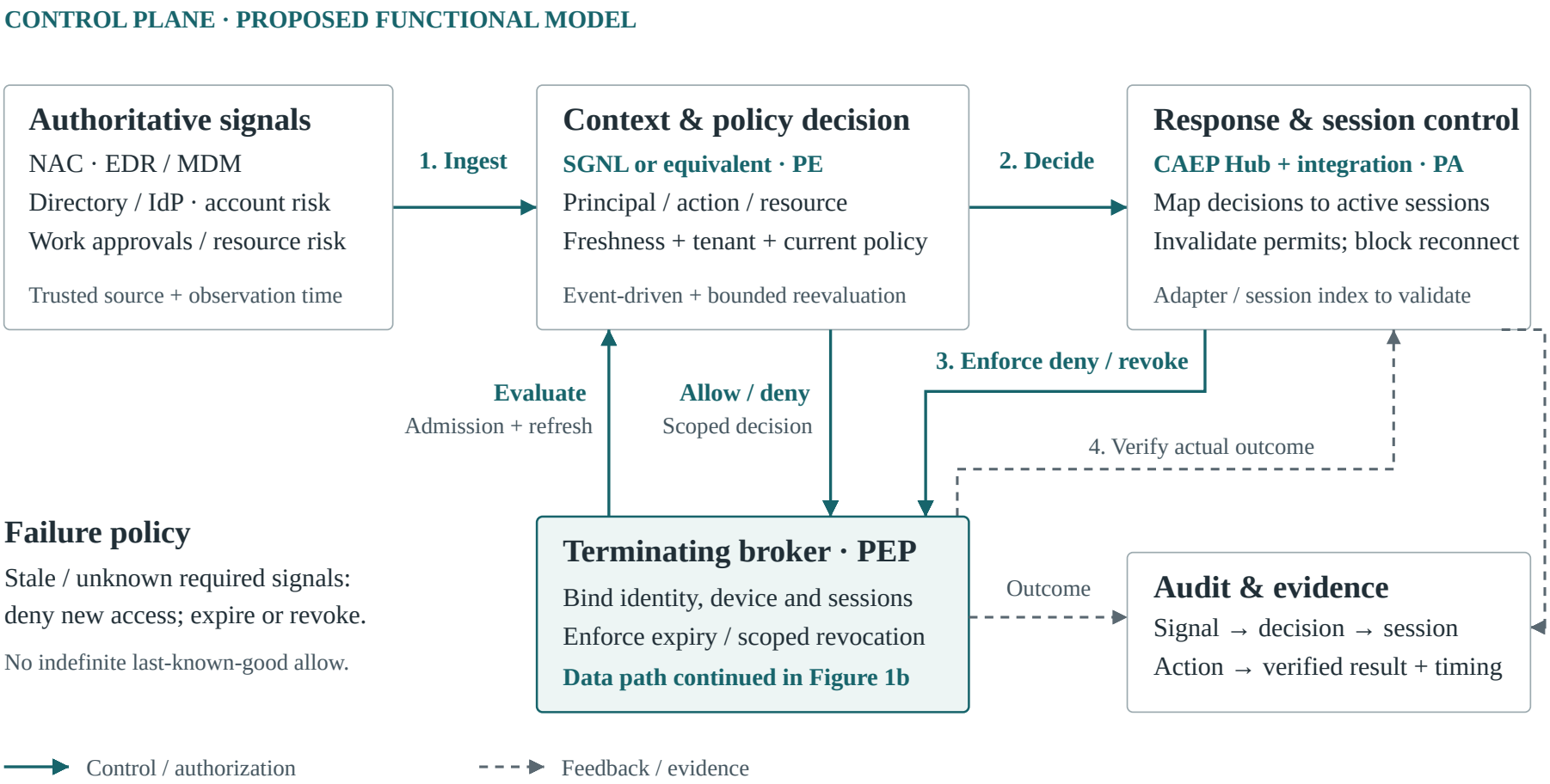
Avoid	Use instead
"VIP/DNS is the tenant identity."	"VIP/DNS selects tenant routing context; validated credentials and policy establish and bind the subject's identity."
"SGNL constantly scans posture."	"Authoritative systems supply posture and context; SGNL or an equivalent service evaluates policy and can orchestrate configured responses."
"OTP re-verifies the session every 30 seconds."	"TOTP is checked at the authentication event; ongoing authorization is separately reevaluated."
"Identity terminates, so enforcement stops."	"Original-subject identity is not propagated to the backend; identity context remains active in the broker's control and audit planes."
"The event was sent, so access is revoked."	"Revocation is complete when the intended enforcement outcome has been verified."
"Flow data shows which user did it."	"Egress flow data attributes traffic to a tenant pool; subject attribution requires the broker's session join."
"The model flagged it, so the session is malicious."	"A behavioral finding is a scored inference; the response follows the approved confidence tier and is explainable from recorded features."

Avoid	Use instead
“The signature is in block mode.”	“The signature’s deployed action has been verified against its approved response tier in configuration and in a live test.”
“Flow analytics shows what the partner did.”	“Flow analytics shows volume, timing, and peers on the encrypted outer flows; operations and content are visible only in the decrypted view at the termination point.”
“No Zero Trust gaps exist.”	“Backend non-propagation is a scoped design boundary; remaining control requirements and residual risks are assessed explicitly.”

Recommended architecture statement: “The brokered-access platform is designed to provide identity-validated, continuously authorized brokered access to legacy resources. Original-subject identity is retained and evaluated within the mediation and control layers rather than propagated into unmodified backend applications. Continued access is conditioned on fresh identity, device/workload, business, and resource context and informed by observed session behavior, with policy changes enforced through scoped denial or session termination at the broker. The strength of this claim depends on verified signal binding, bounded freshness, complete mediation, least-privileged backend credentials, and auditable enforcement outcomes.”

Until the continuous path passes acceptance, use the narrower statement: “The platform provides the brokered-access controls described in the baseline design; continuous contextual authorization and live-session revocation are proposed extensions under validation.” This avoids retroactively attributing the target state to the currently described implementation.

Signal-to-Broker Enforcement Model



PE: policy engine · PA: policy administrator · PEP: policy enforcement point. A delivered event is not proof of terminated access.

Figure 1a. Signal-to-broker control, authorization, and verification loop.

Interpretation: solid arrows carry control requests and decisions; dashed arrows carry outcomes and evidence. The proposed PE/PA/PEP mapping and SGNL integration capabilities remain subject to acceptance testing.

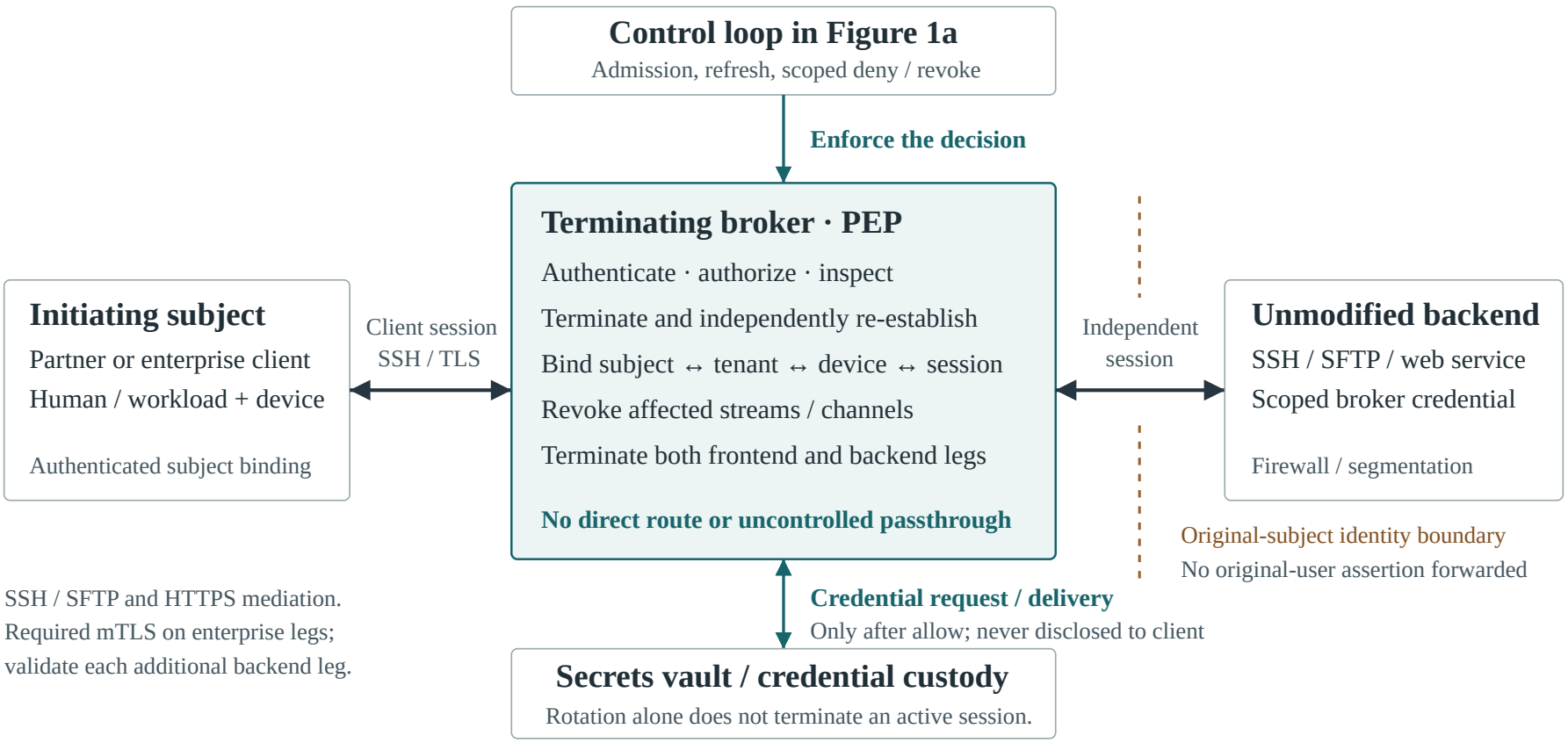


Figure 1b. Mediated data path, credential custody, and original-subject identity boundary.

Interpretation: the broker establishes independent client and backend sessions and applies the decisions in Figure 1a. The backend receives a scoped broker credential, not an original-user identity assertion; terminating transport does not necessarily cancel detached backend work.

APPENDIX: THE ARGUMENT IN NARRATIVE FORM

The following passage restates the argument for readers who want the position without the control detail. It describes the target design, not controls already deployed.

The broker is not the backend, and an allow decision is not permanent

The useful questions are whether the initiating identity is reliable, whether access is authorized, and whether the broker can still enforce that answer when conditions change. They do not require every legacy application to consume the original identity.

NIST's gateway and portal models allow enforcement outside the protected application, but the same architecture also calls for dynamic policy and ongoing reassessment. These are not competing ideas. They describe where control resides and how it behaves over the life of an interaction [1].

A broker can hold the backend credential and establish an independent session without sending the original identity into an application that cannot consume it. The identity must nevertheless remain in the broker's authorization and audit context, and connection-time permission must remain revocable.

A device may become non-compliant, an account may be disabled, or a business approval may end. These changes must trigger a decision linked to the affected sessions. When access is no longer permitted, the broker must act and verify that it stopped.

SGNL's evaluation service returns principal/action/asset decisions, while configured CAEP Hub actions can carry responses to protected systems. Authoritative posture sources and a working broker enforcement interface are still required; a decision or signal alone is not an end-to-end result [5, 6].

The network terminology also matters. DNS and VIP select service context; credentials establish the subject; policy binds the subject to tenant and resource; SNAT distinguishes downstream traffic. These controls are not interchangeable forms of identity.

The limitations remain real. Continuous authorization cannot supply business-action controls the broker does not implement, and transport termination is not necessarily backend job cancellation. Those limitations require compensating controls or explicit risk acceptance, not claims of completeness.

The backend can remain identity-opaque while the access path becomes continuously governed. The broker need not become the backend; it must remain accountable for the access it mediates, including when that access must end.

REFERENCES

[1] NIST SP 800-207, Zero Trust Architecture (2020), §§2.1, 2.2, 3, 3.2, and 3.3. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

[2] IETF RFC 4252, The Secure Shell (SSH) Authentication Protocol, §§5 and 7–9. https://datatracker.ietf.org/doc/html/rfc4252

[3] IETF RFC 6238, TOTP: Time-Based One-Time Password Algorithm, §§3–6. https://datatracker.ietf.org/doc/html/rfc6238

[4] IETF RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3, §§1 and 4.4. https://datatracker.ietf.org/doc/html/rfc8446

[5] SGNL, Access Evaluations, documentation reviewed October 1, 2026. https://help.sgnl.ai/articles/access-service/access-evaluations/

[6] SGNL, CAEP Hub Actions, documentation reviewed October 1, 2026. https://help.sgnl.ai/articles/caep-hub/actions/

[7] SGNL, CAEP Hub Triggers, documentation reviewed October 1, 2026. https://help.sgnl.ai/articles/caep-hub/triggers/

[8] SGNL, Creating and Configuring an SSF CAEP Event System of Record, reviewed October 1, 2026. https://help.sgnl.ai/articles/systems-of-record/creating-and-configuring-ssfevents-caep/

[9] OpenID Foundation, OpenID Continuous Access Evaluation Profile 1.0 (Final), August 29, 2025. https://openid.net/specs/openid-caep-1_0-final.html

[10] IETF RFC 5470, Architecture for IP Flow Information Export. https://datatracker.ietf.org/doc/html/rfc5470

[11] MITRE, ATT&CK Enterprise Matrix, technique reference reviewed October 2, 2026. https://attack.mitre.org/techniques/enterprise/