Nick Clark · Published July 29, 2026 · PDF

Cloud SLA Credit Claims and the Cost of Asserting Downtime

A cloud service level agreement gives the customer a way to assert that the platform failed and to claim a credit for it. Filing that assertion is cheap for the customer, while assessing it costs the provider a review queue, so the two sides of one exchange are priced very differently. U.S. Provisional Application No. 64/117,812 discloses an assertion-cost counter and an authorization budget that meter the asserting party for the act of asserting, epoch-clocked against a hash chain the asserter cannot predict, barred from replay, and verified at the point of receipt. The cost applies without any adjudication of whether the outage claim was right. What changes is that assertion volume becomes a governed quantity rather than a free one.

When filing an outage claim costs nothing

A cloud service level agreement is a promise with a remedy attached. The provider commits to an availability figure, and when the platform falls short the customer may assert that a qualifying outage occurred and claim a service credit. The asserting side of that exchange is cheap: a customer, or a monitoring agent acting for one, notices elevated error rates, drafts a claim, attaches whatever telemetry it holds, and sends it.

The assessing side is expensive. Someone at the provider correlates the claim against internal availability data, decides whether the degradation fell inside covered scope, and either grants the credit or explains why not.

That imbalance shapes behavior in both directions, and neither direction is healthy. Where a credit process attaches no penalty to a claim that turns out to be wrong, the worst case for a claimant is a denial. Where a customer runs automated observability tooling that can generate claims at machine speed, the incentive points toward filing on every anomaly and letting the provider sort it out. A provider's available responses tend to sit on the assessment side: narrower claim windows, heavier evidence requirements, tightly written scope definitions. The honest outage report then arrives in the same queue as the speculative ones and meets the same skepticism.

The structural fact underneath is simple. A credit claim writes an adverse record into someone else's governance state, unpriced for the party writing and costly for the party receiving, and any process built on that asymmetry ends up regulating the receiving side.

Why claim windows and fee caps stall

The responses a contract can reach for are largely recipient-side rationing, and they degrade the signal they are meant to filter. A filing window limits when a claim may be made, not how many. Evidence requirements raise the drafting burden, but drafting is the part automation makes cheap. Per-claim caps limit the provider's payout exposure without touching the volume of assessment work.

Reputation scoring is the obvious alternative, and it fails for a reason worth naming: scoring requires adjudicating merit. Someone must decide a claim was frivolous before the claimant's score can move, which puts the counting of claims downstream of the very dispute the counting was meant to make manageable. The provider becomes both

defendant and judge of whether it should have been accused. Escrow and bonding fare no better, reintroducing merit adjudication as the forfeiture trigger and denominating the stake in a currency unrelated to what the claimant actually does.

What none reaches is the act of assertion itself, priced from a quantity the asserting party needs for its own operation, before any question of who was right.

Inside the assertion-cost counter

Chapter 4 of the filed provisional governs a semantic agent (100) in the role of asserting party; separate chapters govern the same agent as evaluated party and are not described here. Chapter 4 sets out that an agent asserting against a counterparty pays a metered cost for the assertion, applied without any adjudication of the merit of the assertion, and that both directions in which an agent writes into another party's governance state are priced from one quantity held by the party acting. Both roles run against a single authorization budget (404).

An assertion here is a conduct evaluation artifact (116): an identifier of the issuing agent, a recorded assertion time, and a conduct descriptor built from the issuer's own append-only lineage field (104). Pricing runs through the assertion-cost counter (400), held in the issuer's memory field (102), whose fields include an issuance accumulator, a per-recipient-class register, a budget floor field, an epoch reference, and a decrement schedule retrieved from the signed policy object (112) in force. The authorization budget (404) it references is a per-agent quantity maintained across the action space, expressly distinguished in the filing from the per-action-class authorization quantity.

To issue, an agent runs an ordered procedure. The issuer computes the origin-equivalence class (200) of the receiving party and consults the per-recipient-class register. Where that class has already contributed an increment within the current metering window, no increment applies. Where it has not, the accumulator increments by one and, responsive to that increment, the authorization budget (404) is

decremented by the amount in the decrement schedule, which the filing requires to be greater than zero. The units of the decrement and the units gating dispatch of the issuer's own actions are identical, and no exchange rate is applied. The issuer then advances its dynamic agent hash chain to a successor epoch and attests, within the artifact, the accumulator state and that epoch identifier.

Metering runs on those successor epochs, not on a wall clock. Each is generated from a prior epoch, an unpredictability contribution, and a volatile salt, so it is not computable in advance by the issuer, not computable at all by anyone else, and drawn from no clock available to the issuer's execution node. Artifacts attesting a common accumulator state carry a common epoch identifier, so a receiver detects the repetition from its counterparty identity record (114).

At receipt, the receiving agent verifies the attested state as a precondition to admitting the artifact to the receiver's admission evaluator (120). The receiver confirms that the attested epoch identifier is a valid successor of one previously recorded for that issuer, and that the attested accumulator state is not less than a state previously recorded. An artifact lacking an attestation, or failing either check, is not admitted: it produces no determination, moves no value of the scoped integrity vector (106), and increments no counter. An issuer that over-splits its classification of receiving parties to reduce its own increments is caught the same way, by recomputation at receipt and a class-splitting divergence record.

Only one procedure restores the authorization budget (404), and it is narrow: exposure to an uncounted origin-equivalence class (200), bounded above by the authorization budget ceiling (408). Neither elapsed time nor expiry of a metering window restores any budget value.

Rewriting the credit claim workflow

Map this onto SLA credits and the workflow changes at its root. The customer's claiming system holds an authorization budget (404) declared in a signed policy object (112) its own principal signed. Issuing a credit claim as a conduct evaluation artifact (116), it computes the origin-equivalence class (200) of the provider claimed against, and where that class has not yet contributed an increment in the current window, the accumulator increments and the budget is decremented. The claim carries an attestation of that state bound to an unpredictable successor epoch.

The provider's intake now verifies before assessing, and that reordering carries the weight. Rather than assessing first and controlling volume afterward through contract terms, the receiver checks the attestation against its counterparty identity record (114) as a precondition to admission, and a claim whose accumulator state has gone backward, or whose epoch identifier is not a valid successor of one already recorded, never reaches the admission evaluator (120).

Notice what the provider is not doing in that check: deciding whether the outage happened, or whether the customer was reasonable to think so. The filing is emphatic that the decrement is not conditioned on merit: an artifact resolved to the accepted determination (122), the rejected determination (124), the not-determinable determination (126), or the not-applicable determination (128) bears one and the same decrement. The provider never becomes judge of whether it should have been accused.

The constraint that does the real work sits one layer down. Because the decrement lands on the same budget that gates the customer's own action dispatch, in identical units with no conversion, assertion volume competes directly with operational capacity. Responsive to the budget satisfying the floor held in the budget floor field, the authorization gate (300) is written to the withheld state (310) for an enumerated set of action classes, the agent enters the non-executing cognitive mode (302) as to those classes, and an escalation record goes to the principal. What is withheld is executing actions of the enumerated classes, not the capacity to issue further artifacts; but while

the budget stands at or below the floor, the agent attaches no attestation to what it issues, and unattested artifacts do not verify at receipt. A claiming system that has spent its budget therefore reaches a state where its principal is notified, its enumerated actions are suspended, and its further claims land inadmissible, with nobody having ruled on a claim.

The per-recipient-class register changes the shape of the incentive, not just its size. A customer filing many claims against one provider in a window pays one increment, because the class contributed once, while a customer whose tooling sprays claims across many distinct providers pays an increment per distinct class. Cost tracks breadth of assertion rather than depth, leaving the customer with a sustained grievance against its primary platform lightly metered. The filing's own illustration, for a policy declaring one unit per increment against a budget of forty, has an issuer reaching sixty parties across sixty distinct classes exhaust the budget at the fortieth increment, the remaining issuances carrying no valid attestation; the same sixty parties in three classes would have been decremented three units. Those figures are illustrative.

Replenishment then runs in an unusual direction: a claimant regains assertion capacity by being subject to assessment itself. The budget is replenished on receipt of an admitted artifact from an origin-equivalence class (200) absent from the replenishment register (402), by the same amount whichever determination that artifact resolves to, and by an amount the filing requires to be smaller than the per-increment decrement. A party that both consumes upstream services and serves downstream customers keeps its claiming capacity alive by accepting exposure to claims against itself. Where the gate was withheld by reason of the budget floor write, and replenishment raises the budget above that floor, the gate returns to the granting state for the enumerated classes and no others, without a principal-resolution object, without an acceptance determination, and with nothing adverse appended against the party whose claim occasioned the replenishment.

The receiver is protected in turn. A verification cap declared in the provider's own signed policy object (112) limits verifications per issuing class per window; beyond it, further artifacts from that class are logged, not verified, not admitted, and append nothing adverse to the issuing party. The provider gets a ceiling on intake work without a way to punish customers.

Integrating this, and what it leaves unsolved

A plausible entry point is the machine-to-machine layer. Where credit claims are filed by automated monitoring and reconciliation systems, those systems can carry an assertion-cost counter (400) and attach attestations without changing how a human account manager escalates a serious incident. Provider-side intake needs a counterparty identity record (114) per claiming customer and the verification step ahead of the existing assessment queue. Both sides need signed policy objects (112) declaring their windows, decrement schedule, floor, replenishment schedule, and ceiling, and the filing leaves those values to the declaring principal.

Several limits are worth stating plainly. The architecture does not decide whether an outage occurred, whether it fell within covered scope, or what credit is owed; an admitted artifact passes to the admission evaluator (120), and separate chapters of the filing govern admission and determination. Chapter 4 governs the cost of asserting and nothing about the truth of the assertion.

It also does not prevent a well-provisioned claimant from filing. A principal declaring a high ceiling and a small decrement has an agent that can assert a great deal, and the architecture will meter and permit it, supplying a governed quantity in place of an unbounded one. Claims filed outside the architecture are untouched: a customer who emails a spreadsheet is not issuing a conduct evaluation artifact (116).

Nor does any of this eliminate the provider's assessment cost; it bounds the volume arriving through the verified channel. The filing does state that the mechanism depends on no voluntary compliance by the issuing agent and on no party other than the two parties to the exchange, so no third-party arbiter is required, though each side still has to run its own machinery correctly.

Disclosure Scope

This article describes subject matter disclosed in Chapter 4, Assertion-Cost Symmetry, of U.S. Provisional Application No. 64/117,812: the assertion-cost counter, the authorization budget, epoch metering and the bar on replay, recipient-side verification, and replenishment. Mechanisms governing the evaluated-party role and admission and determination are addressed by separate chapters of that filing and are not described here.

The cloud and SaaS service level agreement setting is an application of the disclosed architecture, not a limitation on it; the architecture is general to any exchange in which one party writes an adverse record into another's governance state.

Values referenced as policy-declared, including metering windows, decrement schedules, budget floors, replenishment schedules, verification caps, and the budget ceiling, are set in the signed policy object in force. The filing specifies their relationships and constraints, not their magnitudes; figures above come from its illustrative example.

The application identified here is pending. This publication establishes a public, timestamped record of the disclosed subject matter and does not assert that any party practices it.

Assertion-Cost Symmetry (/assertion-cost-symmetry)

All 49 steps → (/inventive-steps)

Making an accusation costs the accuser too — symmetric, metered, replay-proof.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Assertion-Cost Symmetry: Making an Accusation Cost the Accuser (/articles/assertion-cost-symmetry/making-an-accusation-cost-the-accuser)

SECONDARY TECHNICAL

- Withheld Delegating Faculty, Preserved Direct Execution (/articles/assertion-cost-symmetry/withheld-delegating-faculty)
- Depth-Propagated Harm Inheritance Across a Delegation Chain (/articles/assertion-cost-symmetry/depth-propagated-harm-inheritance)
- Co-Signature Conditioning of Restoration Authority (/articles/assertion-cost-symmetry/co-signature-conditioned-restoration)
- The Anti-Evasion Charge on Governance-Address Removal (/articles/assertion-cost-symmetry/anti-evasion-governance-address-removal)
- The Progressive Issuance Decrement (/articles/assertion-cost-symmetry/progressive-issuance-decrement)
- Non-Attenuable Forward Restraint Inheritance: A Delegation Record That Cannot Strip an Agent's Abstention State (/articles/assertion-cost-symmetry/non-attenuable-forward-restraint-inheritance)
- Holding Authorization When the Renewal Register Cannot Be Read: Re-Read Cadence, Held-Value Resume, and the Max-Hold Inquiry (/articles/assertion-cost-symmetry/register-unavailable-re-read-cadence-held)

APPLICATIONS · GENERAL

- Report Abuse and Takedown Flooding: Making the Accusation Cost the Accuser (/articles/assertion-cost-symmetry/takedown-and-report-abuse)
- The Economics of Agent-to-Agent Disputes (/articles/assertion-cost-symmetry/agent-to-agent-dispute-economics)
- **Cloud SLA Credit Claims and the Cost of Asserting Downtime (/articles/assertion-cost-symmetry/cloud-sla-credit-claims)**

APPLICATIONS · SPECIFIC

- DMCA Notice and Takedown: Where the Cost of an Assertion Falls (/articles/assertion-cost-symmetry/dmca-notice-and-takedown)
- Report Abuse at Scale: Making an Accusation Cost the Accuser (/articles/assertion-cost-symmetry/platform-abuse-reporting)

TERMINOLOGY

- acknowledgment artifact (/glossary#acknowledgment-artifact)
- assertion-cost counter (/glossary#assertion-cost-counter)
- authorization budget (/glossary#authorization-budget)
- dynamic agent hash chain (/glossary#dynamic-agent-hash-chain)

Assertion-Cost Symmetry overview → (/assertion-cost-symmetry)