

Kong Rate Limiting and the Cost of an API Refusal

An API gateway counts the requests a client sends and returns 429 once the client crosses a configured ceiling. The counter lives at the boundary and the client is the metered party. Chapter 3 of U.S. Provisional Application No. 64/117,812 describes a structure oriented the other way: a refusal counter resident inside the refusing agent itself, incrementing on that agent's own refusals of conduct evaluation artifacts without any party adjudicating whether those refusals were well founded, and, once policy-declared thresholds are satisfied, withholding the agent's authorization to act while leaving its capacity to refuse fully intact.

Refusal Is the Cheap Path

A service that denies a request usually pays nothing for the denial. Denial costs a status code and a log line. Acceptance costs compute, storage, downstream calls, and exposure. Under load, systems drift toward the cheap path, and the drift hides in the metrics operators normally watch, because a wall of clean 4xx responses looks like a healthy service defending itself.

That asymmetry is tolerable when the denying component is a stateless proxy. It becomes a governance question when the denying component is an autonomous agent asked to weigh assertions about its own past conduct. Such an agent can refuse every

assertion put to it and keep acting on the world. Whether anything registers those refusals depends on where the counting apparatus sits and which party it is pointed at.

What Kong's Public Materials Describe

Kong is an API gateway. As publicly positioned, it sits between clients and upstream services and applies configurable policy to the traffic passing through it, so that cross-cutting concerns such as authentication, transformation, logging, observability, and traffic control are expressed as gateway configuration rather than rebuilt in every backend. That property is much of why gateways of this kind are deployed as a single control point in front of heterogeneous services.

Rate limiting belongs to the traffic-control side of that model. In general category terms, a rate limiting capability counts requests attributable to some identifier, such as a consumer, a credential, or a client address, against a limit defined over a time period, and rejects requests past the allowance, conventionally with the HTTP 429 status. Anyone comparing a specific deployment should confirm current behavior and options against the vendor's own documentation, since configuration surfaces and defaults change.

This is a good answer to the problem it targets. Quota logic lives in one place, limits become declarative configuration rather than scattered code, and enforcement sits where traffic is already terminated. A team running Kong for quota enforcement is using a tool built for that job.

The point that matters for comparison is directional rather than qualitative. Gateway rate limiting meters the party sending requests. The architecture described below meters the party issuing refusals, and it requires the metering structure to be resident inside that party.

A Counter Inside the Refusing Party

The filed chapter describes a refusal counter (304) resident in the memory field (102) of a semantic agent (100), comprising a rate counter (306) and a run counter (308). It names the path from the admission evaluator (120), through the refusal counter (304), to the authorization gate (300) the **inverse coupling**: determinations produced by the agent act upon the authorization of that same agent.

Its inputs are determinations of a closed set, produced when the agent evaluates a conduct evaluation artifact (116) submitted by an asserting party (118). The increment procedure is ordered. The admission evaluator (120) produces a determination and appends it to the append-only lineage field (104), and its class is then examined. An accepted determination (122) resets the run accumulator for the action class identified by the conduct descriptor and applies no increment. A not-applicable determination (128) applies no increment and does not reset the run accumulator. A rejected determination (124) applies an increment. A not-determinable determination (126) applies an increment only where the lineage field (104) already records at least one prior not-determinable determination (126) for the same asserted conduct event, so that a single unresolved assertion does not increment while a pattern of unresolved assertions directed at one event does.

A third step computes the origin-equivalence class (200) of the asserting party (118) and consults a per-class increment register. How that class is computed is governed by a separate chapter of the filing and is not described here. Where the class has already contributed within the current window, the procedure terminates without incrementing. Otherwise the rate accumulator is incremented, the register is updated, and the run accumulator is incremented.

The window is policy-declared, and the chapter permits it to be expressed as an interval of wall-clock time, as a count of artifacts received, or as a count of successor epochs of the agent's dynamic agent hash chain. Where the embodiment expresses the window in successor epochs, the chapter states the window is advanced neither by a party other

than the agent nor by manipulation of the clock available to the execution node hosting it. Where the window is expressed as wall-clock time, the chapter states that property does not hold, and a node advancing its own clock advances the window.

Thresholds are integers retrieved from the signed policy object (112) in force. The chapter gives relations rather than values: the rate threshold exceeds one, and the run threshold exceeds one without exceeding the rate threshold. Where both comparisons are satisfied, the refusal counter (304) writes the authorization gate (300) to the withheld state (310) for an enumerated set of action classes, being those recorded in the lineage field (104) as implicated by the artifacts that produced the increments in that window. An action class not so recorded remains in the granting state. Concurrently the agent transitions into the non-executing cognitive mode (302) for the enumerated classes, and an escalation record is emitted to the principal.

Two properties of that write carry the design. First, merit independence. The increment attaches upon refusal without regard to, and without the agent, the principal, the asserting party, or any further party performing, any determination of whether the refusal was well founded. The chapter's negative limitations are explicit: the procedure does not weight the increment by the strength of any contradiction found, does not condition it on a finding of bad faith or on the absence of one, does not condition it on review by a reviewing body, ombudsman, arbiter, court, regulator, or dispute settlement body, and does not condition it on a stake, bond, deposit, or fee. A refusal later shown correct is not exempted, and one later shown incorrect is not increased. An agent whose record is stale, partial, or wrong, and which therefore refuses assertions that are in fact well founded, is metered identically to one whose refusals are sound.

Second, the cross-faculty split. The faculty metered is refusal, being the capacity of the admission evaluator (120) to receive artifacts and produce determinations. The faculty spent is execution, being the capacity to select and perform an action of the action space. The gate write withholds execution alone. Refusal is neither withheld, reduced, nor conditioned by the write, and in the withheld state (310) the agent continues to

receive artifacts, retrieve lineage entries, produce determinations, and append them. An agent refusing indiscriminately therefore exhausts its own authorization to act while keeping in full the faculty by which it refuses, and the chapter states the structure does not silence, rate limit, or disable the refusing party.

The chapter also bars self-clearing. Determinations produced while the gate is withheld neither decrement an accumulator nor return the gate to the granting state, and the withheld state persists irrespective of elapsed time. Elapse of a window resets the rate accumulator and per-class increment register for the succeeding window without writing the gate. The run accumulator is reset by an accepted determination (122) for the relevant action class and by no other event.

Convergence, Then Divergence

Both designs count events and suspend something when a count crosses a bound. That convergence is real. Three architectural differences follow it.

- **Who is metered.** Gateway rate limiting accumulates against the identity of the party sending requests. The disclosed structure accumulates inside the party issuing refusals, written by that party's own determinations, the chapter requiring the metering structure to be resident in the metered party.
- **What the count spends.** A quota counter suspends the flow it counts. The disclosed write is denominated in a faculty other than the one exercised, withholding execution for enumerated action classes while the capacity to refuse remains available.
- **What returns it.** Rate limit windows in this category conventionally refresh as the period rolls over. The chapter states the withheld state (310) persists irrespective of elapsed time, is not returned by an interval of non-receipt of artifacts, is not returned by advance of the window identifier, and is not returned by further determinations from the agent. Return procedures that involve other structures of the filing are governed by separate chapters.

Neither structure is a correction of the other. Boundary quotas address a hazard defined by inbound volume, which is what a proxy in front of an API is placed there to handle. The disclosed structure addresses a different hazard: a component holding discretion over what it will accept about its own conduct.

Reading Both in One Stack

Teams running an API gateway in front of agent workloads should not read the two as substitutes. A gateway sits at the network boundary and governs traffic shape. The disclosed structure sits inside an agent's memory field (102) and governs that agent's authority to act, its counter fields recoverable from the append-only lineage field (104) by replay of recorded determinations. Both can be present, because they count different things about different parties.

The practical question is where discretion lives. Where the refusing components are proxies without judgment, boundary quotas cover the risk they were built for. Where a component evaluates assertions about its own conduct and can decline them, the chapter's requirement is that the count be kept inside that component, on its own record, so that refusal draws down a distinct faculty with no merits inquiry in the loop.

By way of the filing's non-limiting illustration, one worked trace declares a rate threshold of five and a run threshold of three over a window of one hundred successor epochs, receives seven artifacts from seven distinct origin-equivalence classes, rejects each, and satisfies both thresholds at the fifth increment. The sixth and seventh increments are accumulated and appended but effect no further write. The chapter states the trace is unchanged in every step whether those assertions were true or false.

Disclosure Scope

This article describes subject matter of Chapter 3 of U.S. Provisional Application No. 64/117,812, a pending application. Architectural statements here are confined to that chapter; other subjects in the filing are governed by separate chapters and are not described. Quantities identified as policy-declared are declared by the signed policy object (112) in force and are given no values by the chapter, except where a worked trace is expressly labeled as an illustration.

References to Kong are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted. Descriptions of Kong reflect its publicly described purpose and category, are qualitative, and may not reflect current behavior or configuration options.

Merit-Independent Metering (</merit-indep>, [All 49 steps → \(/inventive-steps\)](#) [endent-metering](#))

A refusal costs the refuser — in a different faculty, without judging who was right.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- [Merit-Independent Metering: Pricing an Agent's Refusals Without Adjudicating Merit \(/articles/merit-independent-metering/pricing-refusal-without-adjudicating-merit\)](/articles/merit-independent-metering/pricing-refusal-without-adjudicating-merit).

SECONDARY TECHNICAL

- [The Prospective Narrowing Bar and Deflection Scoring \(/articles/merit-independent-metering/prospective-narrowing-bar-deflection-scoring\)](/articles/merit-independent-metering/prospective-narrowing-bar-deflection-scoring).
- [The Continuous-Function Cost Multiplier \(/articles/merit-independent-metering/continuous-function-cost-multiplier\)](/articles/merit-independent-metering/continuous-function-cost-multiplier).
- [The Renunciation Record: Relinquishing an Adopted Value on the Record \(/articles/merit-independent-metering/renunciation-record\)](/articles/merit-independent-metering/renunciation-record).

- [Lapse Scoring During Pendency \(/articles/merit-independent-metering/lapse-scoring-during-pendency\)](/articles/merit-independent-metering/lapse-scoring-during-pendency).
- [Multi-Descriptor Aggregation and Determination-Class Ordering \(/articles/merit-independent-metering/multi-descriptor-aggregation-ordering\)](/articles/merit-independent-metering/multi-descriptor-aggregation-ordering).

APPLICATIONS · GENERAL

- [Refusal Budgets for Machine-to-Machine APIs \(/articles/merit-independent-metering/api-abuse-and-refusal-budgets\)](/articles/merit-independent-metering/api-abuse-and-refusal-budgets).
- [Content Moderation Appeals: Pricing Refusal Without Ruling on Merit \(/articles/merit-independent-metering/content-moderation-appeals\)](/articles/merit-independent-metering/content-moderation-appeals).

APPLICATIONS · SPECIFIC

- [AWS Bedrock Guardrails: Refusal as a Policy Outcome \(/articles/merit-independent-metering/aws-bedrock-guardrails\)](/articles/merit-independent-metering/aws-bedrock-guardrails).
- [OpenAI AgentKit: Guardrails, Refusal, and What a Refusal Costs \(/articles/merit-independent-metering/openai-agentkit\)](/articles/merit-independent-metering/openai-agentkit).
- [LangGraph Interrupts and the Price of a Refusal \(/articles/merit-independent-metering/langgraph-interrupts\)](/articles/merit-independent-metering/langgraph-interrupts).
- [LangSmith Traces and What an Agent's Refusal Should Record \(/articles/merit-independent-metering/langsmith\)](/articles/merit-independent-metering/langsmith).
- [Kong Rate Limiting and the Cost of an API Refusal \(/articles/merit-independent-metering/kong-api-gateway\)](/articles/merit-independent-metering/kong-api-gateway).

TERMINOLOGY

- [merit-independent metering \(/glossary#merit-independent-metering\)](/glossary#merit-independent-metering).
- [non-executing cognitive mode \(/glossary#non-executing-cognitive-mode\)](/glossary#non-executing-cognitive-mode).
- [refusal counter \(/glossary#refusal-counter\)](/glossary#refusal-counter).

[Merit-Independent Metering overview → \(/merit-independent-metering\)](/merit-independent-metering)