Nick Clark · Published July 29, 2026 · PDF

Cloudflare Bot Management: Blocking Agents and Metering Sources

One adversary presenting forty freshly minted identities can carry an agent's refusal rate to its threshold while presenting nothing an authentication check would reject, because every identity may be perfectly well formed. The filed specification treats this as a metering condition rather than an authentication one: the counter charges per assertion, while the contribution that needs bounding is per source. Edge bot management, as publicly described, addresses a related pressure at the network boundary by classifying how likely inbound traffic is to be automated and letting the operator decide what reaches the origin. The disclosed architecture places the same pressure somewhere else: a semantic agent derives origin-equivalence classes from records it already holds, and charges its refusal counter per class.

Forty Identities, One Adversary

An agent that evaluates conduct assertions from other agents has to decide when it has heard enough adverse signal to stop acting. The natural design is a counter: increment on each refusal, compare against a threshold, withhold the action once the threshold is met. That works cleanly against a single counterparty behaving badly, and fails against a counterparty that can mint counterparties.

The filed specification describes the failure precisely. Merit-independent metering accumulates increments on both refusal paths: a conduct evaluation artifact (116) that is false produces the rejected determination (124) and increments, and an artifact unresolvable against the append-only lineage field (104) produces the not-determinable determination (126) and also increments. Volume alone therefore drives the authorization gate (300) to the withheld state (310), and the volume available to an adverse party is limited only by the cost of presenting further identities.

Read that ceiling carefully, because it is the whole of the exposure. An adverse party need not defeat a signature check; the artifacts it presents may each be well formed and each still resolve to a determination that increments. It needs only volume, priced at whatever a further identity costs. The filed procedure responds by changing what the counter charges for rather than by tightening what the agent will accept.

What Edge Bot Management Is Built To Do

Cloudflare bot management, as publicly described, works on a version of this pressure at a different layer, and deserves description on its own terms before any contrast.

Public materials describe it as a capability of a reverse-proxy network positioned in front of customer origin servers, classifying inbound traffic according to how likely it is to be automated. The classification is described as surfaced through a score and through fields usable by firewall and routing rules, so the operator chooses what happens to a given class of traffic rather than having the outcome fixed for them. Those materials also describe a notion of verified or known-good automated clients, letting operators distinguish crawlers and services they want to admit from traffic they do not.

Position is what gives that layer its leverage. A network fronting many properties occupies a vantage no single origin occupies, and that breadth is structurally unavailable to any mechanism confined to one participant's own records. Readers

should consult the vendor's documentation for how the classification is produced; the point here concerns the layer, not the internals.

The question being answered is one of admission: whether a connection reaches an origin, decided on the traffic itself. The disclosed architecture takes up a question beginning after admission, namely how a participant already conversing with a counterparty should account for what that counterparty asserts.

Charging The Counter Per Source

The disclosed procedure forecloses the identity-splitting condition by charging the refusal counter (304) per source of assertion rather than per assertion. A source, in the specification's definition, is a set of asserting parties (118) between which the semantic agent (100) demonstrates a relation from its own records.

Derivation is ordered. On receipt of a conduct evaluation artifact (116), the agent retrieves the counterparty identity record (114) of the asserting party (118), or instantiates one where none exists, then retrieves from the signed policy object (112) in force an enumeration of declared relation types, each specifying a class of entry of the append-only lineage field (104) and a matching condition over such entries. For each asserting party (118) already assigned to a class, in the current window or a preceding one, the agent evaluates each declared relation type against the present party.

Three are named:

- A **shared dispatch lineage (202)** is evidenced where the lineage field contains an entry recording a dispatch to the present asserting party and an entry recording a dispatch to the compared party, and both record a common parent dispatch entry as their immediate antecedent.
- A **co-signature (204)** is evidenced where a single lineage entry bears a signature verifiable against an identity primitive of the present asserting party and a signature verifiable against an identity primitive of the compared party.

- A **common introduction path (206)** is evidenced where the counterparty identity records of both parties record an introducing party and the recorded introducing parties are identical.

Each declared relation type is independently sufficient. Where any is evidenced, the present party joins the class of the compared party; where more than one class is identified, the classes are merged; where none is evidenced, a new class is formed. The assignment, the relation types evaluated, the entries relied upon, and the resulting class identifier are appended to the lineage field, and the identifier is written into each member's counterparty identity record. The specification is explicit that the identifier records class membership and is not a determination concerning that party's conduct.

Aggregation follows. The per-class increment register records whether a class has contributed within the current window, and the increment procedure consults it before touching the rate accumulator, terminating without incrementing where the class has already contributed. An artifact from a party in an already-recorded class still produces a determination and is still appended to the lineage field; it applies no increment. Normalization, per the filing, suppresses no determination and withholds no adjudication, and governs the metering alone.

Locality is the property that governs every comparison drawn below. A second semantic agent holding a different lineage field and different counterparty identity records derives, from the same population, a partition that need not agree with the first. No procedure reconciles the two, no identifier passes between them, and neither agent admits an identifier derived by the other. The class identifier is therefore no identity attested by a third party and confers no portable standing.

Where a party presents identities the agent has never encountered, none of the three relation types can be evidenced from records the agent does not hold, and each such identity is assigned to a distinct class. The filing addresses that case through a severance-survival test over constituent edges. Where every constituent edge of a class resolves not-typeable, the class is designated an untested class, and where such a class's

recorded introduction paths also converge on a common ancestor entry within a depth declared in the signed policy object, a cost multiplier declared in that object applies to the class's contribution. Both conditions have to hold: an untested class whose introduction paths do not converge contributes at full weight notwithstanding the designation. The multiplier is greater than zero and less than unity, and the filing fixes no value for it, illustrating the arithmetic with a declared value of one fifth. A reduced contribution is added, as a rational quantity, to a weighted contribution register rather than to the integer rate accumulator, and the designation lapses as a class accumulates recorded severance events.

Two Layers, Two Different Questions

Both approaches take up the leverage that cheap identity creation gives an adverse party, and part company on placement, on admissible evidence, and on what each one emits.

Placement is the first parting. Network-layer bot management sits in the path, in front of an origin, acting on connections. Origin-equivalence normalization sits inside the deciding agent, acting on assertions already evaluated and already determined.

Evidence is the second. A network vantage point can draw on what it sees across the properties it fronts. The disclosed mechanism is constrained by its own terms to something narrower: evaluation runs on entries of the agent's own append-only lineage field and on counterparty identity records it holds, without reference to a centralized registry, without query to a directory, without participation in a consensus procedure, and without coordination with a further execution node. Those are architectural commitments of the disclosure, not deficiencies attributed to any product.

Output is the third parting, and the sharpest. A traffic classification is a judgment about a client that an operator can act on. An origin-equivalence class identifier is not a judgment. The filing states that it records class membership, passes between no two

agents, and is admitted by no agent other than the one deriving it. Reputation systems generally aim at a verdict that travels; this construction requires one that cannot.

Positioning the two against each other misreads both. Metering per source says nothing about keeping unwanted traffic off an origin, and the disclosed metering assumes a counterparty already admitted. They are complementary by layer.

Coexisting, And What Normalization Does Not Do

Consider an operator running agents that accept conduct evaluation artifacts from peers on an open network. A network-layer control fronts the endpoint and does the admission work, so what reaches the application is traffic the operator has decided to converse with.

The metering condition begins after that point. The party described in the filed scenario presents well-formed artifacts from well-formed identities, its leverage coming entirely from the count charged against the agent's own gate. The disclosed mechanism operates on that population, charging the refusal counter once per class per window.

What the disclosure claims for itself is narrow, and stating it plainly matters more than stating it favorably. The filing describes a bound, not a detector. It bounds the contribution of a single origin to the withholding of an action irrespective of how many identities that origin presents, and only where a declared relation type is evidenced from records the agent already holds. Identities leaving no qualifying lineage entry and recording no identical introducing party are, on the filing's own terms, assigned to distinct classes.

The construction also requires that no registry be consulted and no further execution node be coordinated with, so an operator gets no view beyond its own agent's records and nothing exportable out the other side. The class identifier confers no portable standing, leaving no shared score to escalate.

Everything else follows the policy in force. Relation types are declared in the signed policy object, and an empty enumeration assigns every asserting party to a distinct class, restoring the original condition exactly. The cost multiplier, the continuation interval, and the introduction-path depth are declared there too, the filing defining how each is used and requiring a depth of at least one.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, Chapter 2, origin-equivalence normalization. The application is pending; no product or implementation claim is made here.

References to Cloudflare bot management are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

Descriptions of the subject reflect publicly documented purpose and category at a qualitative level; consult the vendor's own materials for any operational decision. Nothing here assesses what any product does or does not implement, the comparison being between what the disclosed architecture requires and what the category addresses. Cloudflare is a trademark of its respective owner.

Origin-Equivalence Normalization (/origin-equivalence) All 49 steps → (/inventive-steps)

Metering that can't be gamed by splitting one accuser into many.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

– Origin-Equivalence Normalization: Metering Refusal Per Source, Not Per Assertion (/articles/origin-equivalence/origin-equivalence-normalization)

SECONDARY TECHNICAL

- Content-Blind Encounter Commitment and the Paired Encounter Receipt (/articles/origin-equivalence/content-blind-encounter-commitment)
- Cross-Receipt Equivocation Detection: Catching an Agent That Tells Two Stories (/articles/origin-equivalence/cross-receipt-equivocation-detection)
- The Untested-Origin Saturation Bound: Pricing a Flood of Freshly Minted Identities (/articles/origin-equivalence/untested-origin-weighted-saturation-bound)
- Counterparty-Side Fork Detection: Recognizing a Split Identity From Outside (/articles/origin-equivalence/counterparty-side-fork-detection)
- The Attestation Origin-Equivalence Gate: An Anti-Collusion Bound (/articles/origin-equivalence/attestation-anti-collusion-bound)
- The Bridging Party and the Deferred Merge: Charging One Windowed Increment Against Every Origin-Equivalence Class a Single Asserting Party Bridges (/articles/origin-equivalence/bridging-party-deferred-merge)
- The Common-Execution-Node Relation Type: Charging Co-Hosted Asserting Parties as One Origin (/articles/origin-equivalence/common-execution-node-relation-type)
- Receiver-Side First-Encounter Budget Decrement: Pricing a Stranger's Arrival Per Presented-Material Origin Class (/articles/origin-equivalence/receiver-side-first-encounter-budget-decrement)
- Mutual First Encounter Without Cross-Budget Coordination: Two Strangers, Two Matched Pairs, Two Separate Budget Decrements (/articles/origin-equivalence/mutual-first-encounter-no-cross-budget)

APPLICATIONS · GENERAL

- Sybil-Resistant Agent Networks: When One Accuser Becomes Many (/articles/origin-equivalence/sybil-resistant-agent-networks)
- Fraud Rings in Autonomous Commerce: Metering Per Source, Not Per Claim (/articles/origin-equivalence/fraud-rings-in-autonomous-commerce)

APPLICATIONS · SPECIFIC

- W3C Decentralized Identifiers and Verifiable Credentials: Verifying an Identifier Versus Metering Its Origin (/articles/origin-equivalence/w3c-decentralized-identifiers)
- Okta Auth for GenAI: Agent Identity and Origin Equivalence (/articles/origin-equivalence/okta-auth-for-genai)
- WorkOS: Agent Identity Infrastructure and Source Metering (/articles/origin-equivalence/workos-agent-identity)
- Entra Agent ID: Directory Identity and Derived Origin Classes (/articles/origin-equivalence/microsoft-entra-agent-id)

- SPIFFE and SPIRE: Workload Identity and Origin Equivalence (/articles/origin-equivalence/spiffe-spire)
- **Cloudflare Bot Management: Blocking Agents and Metering Sources (/articles/origin-equivalence/cloudflare-bot-management)**

TERMINOLOGY

- asserting party (/glossary#asserting-party)
- origin-equivalence class (/glossary#origin-equivalence-class)

Origin-Equivalence Normalization overview → (/origin-equivalence)