Splats as the Unit of AI Process

Why the token is the wrong unit for agentic work — and what replaces it

May 2026 · this edition July 2026

Abstract

Every major foundation-model API bills, displays, and rate-limits in tokens — substrings of a vocabulary, content-blind to the work done on the user's behalf. A user paying for tokens is paying for string surface area; the relationship between tokens and useful work is opaque. This paper argues for a different primitive: the **splat**, the smallest complete unit of agent work. Mathematically, a splat is a weighted anisotropic Gaussian — a (center, covariance, amplitude) triple drawn from the splat-regression literature. Operationally, it is a hash-chained action receipt: an input, the action it resolved to, the diff it produced, and the approval that let it through. Doctrinally, it is the atomic injection of encouragement — a completed Verify-to-Backpass cycle of measured work. Tokens compress language surface; splats compress learning events. We develop the definition, state the compression claim, trace the implications for display, billing, rate-limiting, and oversight, and position the unit against three recent frameworks (Bennett 2025, MemOS 2025, AEX 2025).

Epistemic legend

This paper preserves the claim-tagging discipline of the Aria Thesis it derives from. Non-trivial claims are marked:

- **[T]** Thesis-known — sourced from CertusOrdo doctrine. Confident.
- **[F]** Field-standard — established in the relevant academic literature. Confident.
- **[I]** Ian-original — an extension or reframe not found in the source literature.
- **[A]** Aria-formulation — a formalization proposed to fill a gap; not authoritative until tethered to source.

Where a claim describes something proposed rather than shipped, the text says so. Where it describes production behavior, the text says that instead — and section 6 states which is which as of this edition.

---

1. The token is the industry primitive

Every major foundation-model API — Anthropic, OpenAI, Google, Meta, Mistral — bills, displays, and rate-limits in **tokens**. **[F]** A token is a substring of the model's vocabulary, an ID in a lookup table, content-blind to the work done on the user's behalf. **[F]**

A user paying for tokens is paying for *string surface area*. The relationship between tokens and *useful work* is opaque: a 1-token answer can be a finished proof; a 10,000-token answer can be a hallucinated wander. **[I]**

This was tolerable when language models were text-completion engines: the text emitted *was* the product, so a unit of text was a defensible unit of account. It stops being tolerable the moment the model becomes an agent. An agent's product is not text — it is actions taken against real systems: files edited, services restarted, records written, approvals exercised. Measuring agentic work in tokens is measuring a construction project by the weight of the conversation on the job site. The industry's unit of account has not kept up with what the systems now do. **[I]**

There is a second, subtler failure. Transcripts — the token-level record of a session — are a poor audit surface. They mix intent with output, they are lossy about ordering, and they say nothing about which actions were actually taken against a real system. An audit that cannot be replayed is a story, not a record. **[I]** A better unit of process must be a better unit of *record* at the same time.

2. The splat: definition

The splat has one identity viewed from three distances: a mathematical form, a doctrinal form, and an operational form. This section gives all three and states why they are the same object.

2.1 Mathematical form

A **splat** is a triple $s = (b, A, v)$ with center $b \in \mathbb{R}^d$, symmetric positive-definite covariance $A \in \mathbb{R}^{d \times d}$, and amplitude $v \in \mathbb{R}$. **[F]** It induces a function

$$\psi_s(x) = v \cdot (\det A)^{(-1/2)} \cdot \exp\left(-\tfrac{1}{2}\,(x-b)^{\top} A^{-1} (x-b)\right)$$

up to a normalization convention. **[F]** A **splat regression model** with k splats is the finite sum

$$f(x) = \Sigma_{i=1..k} \psi_{s_i}(x)$$

with parameter set `θ = {(b_i, A_i, v_i)}`. **[F]** Training follows the Wasserstein–Fisher–Rao (WFR) gradient flow on signed measures: moving a splat across the input space is *transport*, growing or shrinking its amplitude is *birth/death*, and the flow induces coupled dynamics on `(b, A, v)`. **[F]** The full mathematical development — including the Cholesky-of-precision parameterization that removes matrix inversion from the training loop — is in the Aria Thesis (v0 §2–§5) and is not repeated here.

The three components have plain-language readings that carry through everything that follows:

- `b` — *where in state-space* the work happened.
- `A` — *how the work was distributed and correlated.*
- `v` — *how much the work weighed in the final answer.*

2.2 Doctrinal form

The CertusOrdo cycle runs `1 Verify → 2 Decide → 3 Correct → 4 Document → 5 Learn → 6 SPLAT (Fulcrum) → 9 Backpass / Release → back to 1`. **[T]** The SPLAT is the position-6 fulcrum where pre-state meets post-state and truth is measured; the Backpass releases the measured truth into the next cycle. **[T]**

The locked doctrinal definition:

> **The splat is the atomic injection of encouragement. [T]**

A regression splat does the same thing the doctrinal splat does: at any input, the splat occupies a region of state-space (centered at `b`, shaped by `A`) and contributes its weighed measurement to the prediction. The forward pass enters the splat; the backward pass exits through it. The mathematical splat `(b, A, v)` and the doctrinal splat (atomic injection of encouragement) are **the same object viewed from two distances** — the fulcrum metaphor is literal, and the use of the same word on both sides of the document is deliberate. **[T][I]**

2.3 Operational form

In production, the splat is a **hash-chained action receipt**: the smallest complete unit of agent work, recorded as an input, the action it resolved to, the diff it produced, and the approval that let it through. Each record is hashed and chained to the one before it, so the sequence cannot be silently edited after the fact. **[I]**

This is the same object again, at engineering distance. One receipt is one completed Verify→Backpass cycle: the input is the pre-state, the diff is the post-state, the fulcrum is where they

met, and the hash chain is the Document step made tamper-evident. A splat is a small, fixed-size record — `d(d+3)/2 + 1` floats in the mathematical form — that compresses an entire cycle of measured work. **[I]**

The chain form is what makes the splat a unit of *record*, not just a unit of *account*. It buys three things a transcript cannot:

- **Replay** — the sequence can be re-run and compared.
- **Attribution** — every change has a decision attached to it.
- **Verification** — a third party can check the chain without trusting the operator.

3. The claim

> *Splats are a smaller unit of measurement than tokens, and carry more information per unit than characters or tokens, because they encode process rather than surface.* **[I]**

The compression argument: a token is `O(log V)` bits of vocabulary ID plus `O(D_emb)` bits of embedding — typically 4–16 KB per token in float16 once the embedding is counted. A splat in `d = 4`–`16` is $O(d^2)$ floats — at `d = 8`, roughly 36 floats, about 144 bytes in float32. **[I][A]**

But the byte counts are the lesser point. The information content of the two units is *incomparable in kind*: **tokens compress language, splats compress learning events. [I]** A token can only encode a substring. A splat carries process information a token cannot: the gradient, the learning delta, the location of the fulcrum — *what the system did, where, and how confidently*. **[I]**

The practical consequence is the inversion of an incentive. When the unit of account is the token, the accounting silently rewards verbosity: more emitted text is more billable surface, whether or not the work advanced. When the unit is the splat, the accounting counts completed cycles of measured work — and a system that solves the problem in fewer cycles registers as cheaper, not smaller. The unit you count is the behavior you select for. **[I]**

4. Implications: display, billing, rate-limiting, oversight

The unit of process determines four surfaces of an agentic product. In each case the splat replaces the token at the user-facing layer of Vox Ordo. **[I]**

What the user sees. The tracker displays **splats burning**, not tokens burning: “12 splats burned this turn — 3 in retrieval, 6 in reasoning, 3 in code generation.” The user immediately reads *how*

much work was done, not *how much text was emitted*. The three-way breakdown is possible precisely because each splat records where in state-space its work happened.

What is billed. Cost per splat replaces cost per token as the user-facing unit of account. Splats correlate to actual effort — cycles completed, learning deltas absorbed — so pricing aligns with value rather than verbosity. A model that solves the problem in fewer splats costs the user less: the opposite incentive to token-billing, which silently rewards verbose models.

What is rate-limited. Splats per second, per session, per project. A misbehaving loop spikes splat rate before it spikes token rate — a runaway agent completes (or aborts) action cycles faster than it inflates its prose — so splats are the earlier signal for throttling and anomaly detection.

What is overseen. The splat chain is the audit surface. Because every action emits a hash-chained receipt, oversight does not depend on a human reading the transcript in real time: a shipped output is acceptable if and only if an auditor *could, in principle*, reverse-engineer it from its splat receipts — replaying the sequence, attributing each change to its approval, and verifying the chain independently. At production latency, where a human cannot evaluate every agent action as it happens, receipt-level audit is the form oversight takes. **[I]**

The unit is already real in the system: the splat tracker and per-splat scoring run in production (see §6). What these four surfaces describe is the same production object surfaced to the user.

5. Positioning against the field

Three recent frameworks touch the same territory from different directions. In each case the splat is not a competitor to the framework's own unit — it sits at a different layer, below or beside it. This section compresses the full overlay analyses (Aria Thesis v2 §8.7–8.9).

5.1 Bennett (2025): a unit below the tool layer

Bennett's AGI framework names two foundational tools — search and approximation — and argues hybrids outperform pure approaches. **[F]** The splat sits *below* that tool layer. A token measures one parameter of approximation; a search-step measures one parameter of search; **a splat measures one completed Verify→Backpass cycle of measured work, regardless of which tool produced the cycle.** Hybrid systems need a unit that hybrid measurement requires; the splat is that unit. **[I]**

5.2 MemOS (Li et al. 2025): the receipt on the memory, not the memory

MemOS builds a memory operating system around the **MemCube** — an atomic unit of stored memory carrying content plus provenance and versioning metadata. The splat is not a MemCube. **The splat is the audit-stamp on the lifecycle event that produced or modified a MemCube.** A MemCube is the unit of stored memory; the splat is the unit of attribution and directional encouragement for the action that stored it. MemCube is the noun; the splat is the verb's audit receipt. Different abstractions, both required — a substrate could adopt MemOS's capabilities verbatim *and* layer splats on top; the two are stackable, not competing. **[I]**

5.3 AEX (open-experiments 2025): the reasoning receipt under the commercial receipt

Agent Exchange is a programmatic marketplace for agent-to-agent work; its atomic unit at the work layer is the **contract**, and its audit layer is the settlement ledger. The settlement ledger is post-hoc: it records that the contract closed, not *which reasoning produced the bid*. The splat operates at the action layer below the contract layer — a single contract produces many splats, one per reasoning step, on both consumer and provider sides. The contract is the *commercial* receipt; the splat chain is the *reasoning* receipt. Both are required for a marketplace whose participants can be audited rather than merely trusted. **[I]**

The pattern across all three: each framework has a real unit at its own layer — the tool, the MemCube, the contract — and each is missing the process-level receipt underneath it. The splat is that receipt.

6. What this is not

- **Not a marketing rename.** The splat is a real object in production code — the splat tracker and per-splat scorer — and a real object in the math (§2). The unit-of-measurement claim is downstream of the object's reality, not upstream of it.
- **Not a proposal.** The unit is live: splat scoring ships in production today, and the system has recorded **61,000+ lifetime splats as of July 2026**. What remains in motion is the display layer — some surfaces in Vox Ordo still show token counts, and the migration of user-facing display from tokens to splats is underway. This paper argues the *why*; the migration itself is an engineering matter, not a research one.
- **Not exclusive of tokens internally.** Where a foundation-model dependency still bills in tokens, the system accounts for it at the boundary and converts to splats for user-facing display. **[I]** The claim is about the right unit for the user-facing layers of agentic work — display, billing, rate-limiting, oversight — not a denial that tokens exist underneath.

Claims in this paper marked **[A]**, and the LOCOMO-class benchmark comparisons of the parent thesis, remain proposals and predictions respectively; they are stated as such and are not production claims.

This edition (July 2026) supersedes the original PDF; brand references updated (the product is Vox Ordo; Aria is the agent). Doctrinal and mathematical content unchanged from Aria Thesis v0 §8 / v2 §8, preserved verbatim at github.com/iansteitz1-eng/aria-thesis.