When Should AI Agents Pay per API Call?

Stablecoin micropayments, metered billing, and delegated spending

Jace Shim · November 2026 · https://doi.org/10.5281/zenodo.24591069

Abstract

An AI agent may need a single data lookup from a vendor it will never use again, or thousands of image-processing calls from the same vendor. These are different billing problems. Drawing on Chapter 13 of Target Money, this paper distinguishes authorization, metering, billing frequency, and settlement. It proposes a break-even comparison between settling each API call, settling batches of metered calls, and buying a subscription. Per-call settlement is most plausible when use is sporadic or sellers lack a reliable way to collect later. At high volumes, a small fixed fee on every payment can outweigh the cost of carrying a capped unpaid balance. A numerical example and a pilot design specify how to test the trade-off.

1. A receipt for every call?

Imagine an agent that builds a report using a one-off company lookup and a thousand page-extraction calls. A subscription to the lookup service may be excessive if a single purchase is allowed. For the extraction service, however, paying for every page can mean repeated settlement costs. Both vendors can still charge by the unit of service. The choice is when to move funds.

Google's Agent Payments Protocol (AP2) illustrates a payment-agnostic approach to recording delegated authority. The x402 white paper describes an HTTP-based approach to requesting and presenting payment for internet resources. These protocols motivate the application. The comparison here asks what each billing arrangement costs when the same service, authority and usage records are held fixed.

2. Four decisions in a usage-based payment

Authorization answers who may buy, from whom, for what purpose and within what budget. Metering determines which events count as billable. Billing frequency determines when charges are grouped. Settlement determines how the vendor receives funds. A system can meter every call while transferring stablecoins only when a recipient-specific amount or time limit is reached.

A buyer can delegate purchases of a specified API up to a budget, set a maximum unit price and define when the permission expires. Each service request carries a unique reference. A contract or service agreement should define whether a completed result, a timed-out request, or a retry is billable. When the result is uncertain, both records can be reconciled before charging again. The budget calculation should cover settled and pending charges, with a provision for known fees. A delayed refund should not restore spendable authority until the accounting records confirm it.

3. A break-even rule for payment frequency

Measure the cost of each option over the same observation window and service quality. Let N be the number of completed billable calls, p their common price, f the fixed cost of one settlement, and B the number of batch settlements. Let L be the expected incremental cost of batch collection delay, nonpayment, and reconciliation relative to per-call settlement. The shared usage charge is $N \times p$. With the same fixed settlement cost on both paths, batching is cheaper when $(N - B) \times f$ exceeds L. If settlement prices, conversion costs, or failure costs differ by path, include them separately. Compare a subscription at its actual price and coverage, including any overages and charges for months with little use.

Suppose a thousand calls cost 0.02 currency units each and each transfer costs 0.004 units. Per-call fees total 4, on top of the 20 usage charge. Ten batch settlements cost 0.04. Batching can absorb up to 3.96 units of incremental delay, collection and reconciliation cost before it loses its fee advantage. These values are illustrative. Where a vendor will not extend credit, a small prepaid balance is an alternative. It shifts the exposure to the buyer, who can limit the amount held and specify how unused funds are returned.

4. A pilot for a paid document-parsing API

Compare per-call settlement, capped batched settlement, and a conventional metered invoice. Fix the price per usable result, task set and access rules. Run a shadow phase using synthetic calls and simulated receipts, then compare real outcomes only within a separately authorized pilot. Include timeouts, disputed receipts and service outages in the test set. A payment receipt can show funds moved and still leave the quality of the output open to review.

Report net cost per usable result, time from request to usable response, wrong or duplicate charges, dispute resolution time, vendor unpaid exposure, and unused buyer balances. Vary the payment rail within the same billing policy where feasible. This separates any advantage of stablecoin settlement from the savings generated by batching. Repeat under bursty and steady workload patterns. The prediction is that recurring high-volume relationships will favor aggregation when collection risk is modest. Sporadic purchases from new vendors may justify per-call payment if access requires immediate settlement and the cost per transfer is small. The paper reports no observed effect estimates.

5. Relationship to earlier papers

Programmable Money in Three Layers separates the asset, purpose and evidence used in conditional settlement. Here the evidence is a billable API event, and the design question is how many such events should be settled together. When Should Freelancers Get Paid? studies milestones that require a substantive review. This paper studies repeated low-value requests with an agreed definition of completion. The programmable-voucher paper evaluates eligibility at checkout. Here the recurring decision is whether an outstanding charge remains within a continuing delegation.

Keywords

Agentic payments, stablecoin micropayments, API billing, metered billing, delegated spending, batch settlement, pay per use.

References

Google Cloud. Announcing Agent Payments Protocol (AP2). https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol

Shim, J. (2026). Target Money, Chapter 13. Raon Book.

Shim, J. (2026). Programmable Money in Three Layers. https://doi.org/10.5281/zenodo.24580931

Shim, J. (2026). When Should Freelancers Get Paid? https://doi.org/10.5281/zenodo.24581270

Shim, J. (2026). When Do Programmable Vouchers Improve Public Spending? https://doi.org/10.5281/zenodo.24578261

x402. The Payment Protocol for Agentic Commerce. https://x402.org/wp-content/uploads/sites/10/2026/06/x402-whitepaper.pdf