

The Duration of Token Demand

What 5,803 Boundless orders reveal about reusable collateral

Nam Beethoven | 13 September 2026

In a 14-day sample of Boundless, 45.24% of locked orders generated just 1.09% of collateral-time. Another 54.11% generated 98.86%. Nearly half the activity occupied almost none of the capital. I reconstructed these exposures from 5,803 orders in the public Base indexer. [1]

My thesis is that growing throughput is insufficient evidence of growing demand for a reusable collateral token. The relevant bridge is collateral-time: each bond multiplied by how long it remains encumbered. A network can become more useful while requiring less collateral per unit of work; even its absolute active bond requirement can fall if reuse accelerates faster than demand grows.

Multicoins' Internet Labor Markets connects increasingly granular work, verification and contributor economics. I extend that framework into a narrower question: when does more work require more tokens to remain committed? Boundless provides a concrete case in which headline activity and capital occupancy describe very different markets. [2]

The clock behind a bond

Boundless matches buyers of cryptographic proofs with provers. A prover can lock a request by reserving collateral from its available market balance. Eligible, timely fulfillment credits that balance again; failure exposes the bond to slashing. The collateral documentation describes a typical bond worth roughly ten times the request's maximum fee. That is a per-request guideline, not a measure of simultaneous capital demand. [3-5]

The distinction matters because the same collateral can support successive obligations. A million tokens committed cumulatively across many orders need not mean a million distinct tokens are unavailable at once.

For comparable jobs in a steady system, let q be new locks per day, b the bond per job, and T its holding time in days. An average of qT obligations are outstanding. Multiplying by their bond gives average active collateral:

$$C = q \times b \times T$$

For 100,000 daily jobs, \$10 bonds and one-hour holding periods, the calculation is $100,000 \times \$10 \times 1/24 = \$41,666.67$. If daily jobs double while holding time falls to ten minutes, it becomes $200,000 \times \$10 \times 10/1,440 = \$13,888.89$. Twice the activity requires one-third the average active collateral. These are hypothetical steady-state examples, with unchanged bond size and successful return.

This is standard stock-flow accounting. Heterogeneous jobs require $q \times E[bT]$, the arrival rate multiplied by the mean of each bond's size-times-duration. Multiplying separate means, $q \times E[b] \times E[T]$, discards the relationship between size and duration. It works only when the omitted covariance is zero. Peak concurrent demand and idle operating reserves remain separate funding requirements.

The job mix changes the answer

I collected a frozen snapshot of locks opened from 27 August through 9 September 2026 UTC. Fourteen API pages contained 7,000 rows. After removing two identical duplicate digests and filtering by lock time, 5,803 distinct requests remained. Their daily lock counts and cumulative bond amounts matched the

same indexer's daily aggregates on all 14 days. This establishes internal consistency; it is not independent receipt-by-receipt validation. [1]

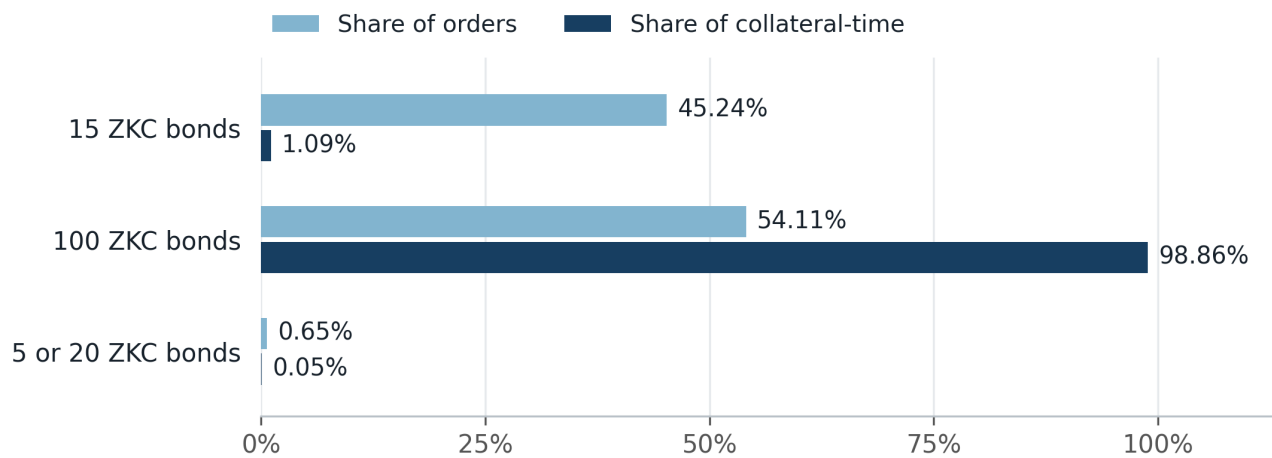


Figure 1. Activity and capital occupancy diverge. Shares of 5,803 locks and their full-lifetime collateral-time, grouped by bond size. The 5 and 20 ZKC groups contain 38 orders in total. Source: preserved indexer responses and original calculations [1].

The 2,625 orders bonded at 15 ZKC held collateral for 4.24 minutes on average. The 3,140 orders bonded at 100 ZKC held it for 48.37 minutes. The larger-bond jobs were also much slower to resolve. Their combination explains why 54.11% of orders consumed 98.86% of collateral-time.

The locks committed 354,075 ZKC cumulatively and accumulated 921,767,640 ZKC-seconds of full-lifetime exposure. Their bond-weighted mean duration was therefore $921,767,640 / (354,075 \times 60) = 43.39$ minutes, versus an unweighted 28.16 minutes. Dividing the full-lifetime exposure by the 14-day arrival window gives 762.04 ZKC of equivalent average exposure.

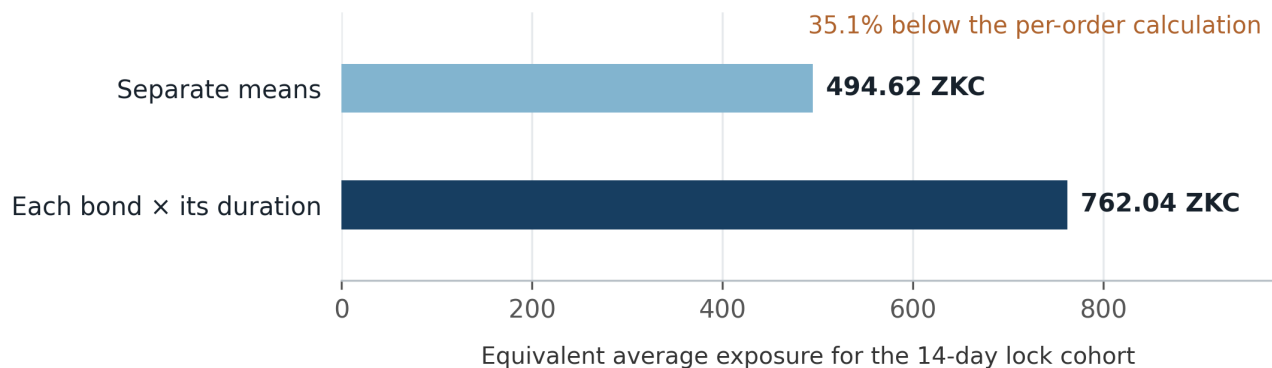


Figure 2. Multiplying separate means understates exposure by 35.1%. Both calculations use the same 14-day lock cohort and full-lifetime durations, including resolutions after the window. The 762.04 ZKC result is a cohort-normalized exposure equivalent, not total market collateral or TVL. Source: original calculations [1].

Using unrounded inputs, the shortcut produces only 494.62 ZKC: $414.5 \text{ locks per day} \times 61.016 \text{ ZKC per bond} \times 28.16 / 1,440 \text{ days}$. It understates the per-order calculation by 35.1%. A model can therefore use the correct average bond, correct average duration and correct activity count, yet materially miss the capital requirement.

The sensitivity is equally uneven. Halving every 15 ZKC order's duration, with everything else unchanged, removes only 0.54% of total collateral-time. Halving the 100 ZKC group's duration removes 49.43%. These are mechanical counterfactuals, not observed upgrades. Improving almost half the jobs can have little capital effect if those jobs barely occupy capital.

The main limit is concentration: two requestor addresses generated 99.1% of locks. Addresses are not necessarily independent customers, and I do not establish whether these workloads are external, affiliated or subsidized. These percentages describe this cohort. The accounting method generalizes; the observed mix does not. A change in who submits work could move the metrics without any improvement in technology.

Fulfillment can leave the original bond outstanding

A request can be fulfilled while its original prover's collateral remains encumbered. If that prover misses its deadline and someone else delivers, the customer's result does not establish repayment to the original prover. I therefore distinguish general fulfillment, timely original-prover delivery and recorded slashing. [1, 5]

One preserved record makes the distinction visible. A prover locked 100 ZKC at 16:05:25 UTC on 28 August. Another prover fulfilled the request at 20:05:17, just after the original lock deadline. Slashing was recorded at 00:06:11 on 29 August. The original obligation lasted eight hours and 46 seconds, about twice the time to customer fulfillment.

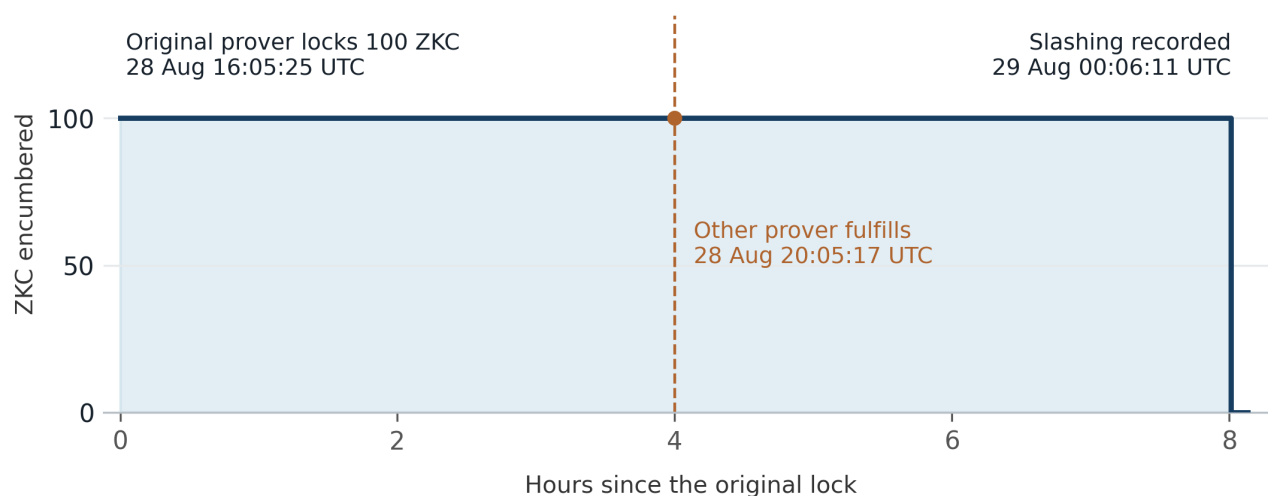


Figure 3. One request, two different endpoints. Reconstructed from requests-010.json, digest 0x7f08e2ac...cfccb. The original-prover delivery field is null and the fulfiller differs from the locker. The drop ends the original encumbrance through slashing; it does not show a refund. The archive preserves identifiers and transaction links [1].

The broader pattern matters more than this example. Of the 5,803 locks, 5,730 had timely original-prover delivery and 73 were slashed; none remained unresolved at collection. Sixty-six slashed requests also had a recorded fulfillment. The slashed group represented just 1.26% of locks but 10.94% of full-lifetime collateral-time.

An apparently strong fulfillment rate is therefore insufficient to characterize prover economics. Failed obligations can occupy disproportionate capital before resolution. Slashing then imposes loss and possible replenishment needs, which belong in a separate model. I do not treat either a timeout or another prover's delivery as automatic collateral return.

Faster proofs matter only if the obligation ends sooner

The relevant clock runs until capital ceases to be encumbered. Faster local computation need not move that endpoint by the same amount. Batching, submission and settlement can intervene; Boundless explicitly batches proofs to amortize verification costs. The useful performance measure is bond-weighted time to eligible release, with failed obligations tracked separately. [4]

Filecoin illustrates the boundary. Its collateral mechanisms include an initial pledge for each storage sector, block rewards that vest over time, and storage deal collateral. The initial pledge helps secure the sector's promised lifetime. A provider can generate recurring proofs faster while still owing the remaining storage service; sector expiration governs release of the associated pledge. [7]

My inference is that faster proof production alone need not shorten that storage obligation. A delivery bond and a pledge supporting continuing availability can respond differently to the same technical improvement. Filecoin is a mechanism comparison here, not a second measured dataset. It prevents an overbroad conclusion that better verification must reduce collateral across every network.

A business selling faster release also needs meaningful unit economics. At an assumed 10% annual capital cost, financing a \$10 bond for one hour costs about \$0.000114. Cutting the hold to ten minutes saves \$0.000095. A one-cent service fee would be roughly 105 times the financing saving.

For a \$1,000 obligation shortened from seven days to one, the same assumed rate implies \$1.64 saved. A \$0.50 service could cover its fee through financing savings if it actually accelerates release without weakening assurance. Both examples are hypothetical. Reduced losses, scarce credit or earlier delivery could add value, but each needs separate evidence; none is silently included in the interest calculation.

Denomination changes the token-demand bridge

All four observed bond sizes are expressed in ZKC. That does not establish a permanent fixed-token policy. Boundless's SDK documentation also describes specifying dollar-equivalent collateral. The unit recorded by the contract and the economic target chosen by the requestor are different things. [4]

If each comparable job requires a fixed k tokens, its steady active-token requirement is qkT . A higher token price increases the dollar capital committed without automatically reducing k . If instead new jobs target B dollars and convert at token price P when posted, the new bond is $\frac{B}{P}$ tokens. With price stable over the modeled interval, the active-token requirement is $\frac{qBT}{P}$.

This distinction rules out a mechanical path from more dollar-valued work to a higher token price. Historical dollar exposure requires contemporaneous prices and the actual bond-setting policy. Existing fixed-token locks are not retroactively resized simply because the price changes.

Active bonds also cover only one holding motive. Idle operating balances, precautionary reserves and staking can dominate them. Boundless's mining framework links reward eligibility and caps to staking, a separate reason to hold ZKC. Reusing a bond sooner does not make those balances disappear. Nor does a new lock necessarily create a market purchase: it may draw on existing inventory. [6]

What I would underwrite

I would require a collateral-demand forecast to preserve the joint distribution of bond size and holding time, identify what terminates each obligation, and distinguish native-token requirements from dollar targets. I would then model reserves, staking and loss replenishment separately. Aggregated order growth cannot substitute for that bridge.

The Boundless cohort demonstrates a measurement failure, not an observed causal decline in network collateral or a fair-value target for ZKC. It contains no controlled comparison around an upgrade. A stronger demand outlook could still be correct if paid workloads grow faster than turnover improves, if large obligations become more common, or if other holding motives dominate.

The investment implication is nonetheless concrete. In this sample, nearly half the orders account for roughly one percent of collateral-time. A thesis built on job growth can correctly anticipate adoption

while overstating the capital that adoption requires. Before treating more work as evidence of more token demand, I want to know which obligations are growing and how long their collateral remains unavailable.

Methods and reproducibility

Scope and provenance. Base, chain ID 8453; market 0xfd152dad5183870710fe54f939eae3ab9f0fe82. Selection is by locked_at in [2026-08-27 00:00 UTC, 2026-09-10 00:00 UTC). Collection ran 10 September, 20:20:41–20:22:43 UTC. The archive preserves 17 raw API responses, retrieval URLs, timestamps and SHA-256 hashes. There are 5,803 unique request digests and IDs, eight requestor addresses and 12 provers. Unlocked fulfillment is outside this lock cohort.

Exposure endpoints. For each lock i , l_i is locked_at and b_i is integer lock_collateral divided by 10^{18} . If lock_prover_delivered_proof_at exists and is no later than lock_end, e_i is that delivery timestamp. Otherwise e_i is recorded slashed_at. No selected lock was unresolved at collection. Timely original-prover delivery is used as the proxy for eligible balance credit under the inspected implementation. Slashing ends the original encumbrance through disposition of the bond, not full repayment.

Window and cohort accounting. Full-lifetime exposure sums $b_i \times (e_i - l_i)$. Dividing by the 14-day arrival window gives 762.043353 ZKC, the exposure equivalent used in Figure 2. Group shares and the 35.1% comparison use the same full-lifetime durations. Clipping every duration to the observation window gives 760.046908 ZKC, the cohort's contribution to actual in-window average active collateral. Positions opened before the window are excluded, so neither is a complete network balance. Below is the clipped calculation for the cohort in window $[S, E)$.

$$C_{cohort} = \frac{\sum_i b_i \max[0, \min(e_i, E) - \max(l_i, S)]}{E - S}$$

Checks and limits. Daily lock counts and cumulative collateral flow reconcile on all 14 days; an event-sweep integration matches the sum of clipped per-lock exposures. These are checks within one indexer. Attempted independent Base RPC receipt checks returned HTTP 403; no complete receipt audit or deployed-bytecode match was obtained. The indexer status snapshot's zero timestamp is not used to prove freshness. Concentrated requestor addresses prevent treating this as evidence of diversified commercial demand or a market-wide trend.

Reproduction. Extract Boundless-Collateral-Evidence.zip and run python3 analyze.py. The standard-library script verifies raw hashes and regenerates cohort.csv, daily.csv, bond-groups.csv and summary.json without network access. Run python3 final-essay/make_figures.py with matplotlib and numpy installed to rebuild the three figures. The archive also contains the exact example record, transaction links, formula checks and source notes. The live API may subsequently change; the archived snapshot is the reference for every empirical number here.

Sources and evidence

Documentation checked 10–12 September 2026. Empirical results refer to the archived 10 September snapshot; hypothetical calculations are labeled in the essay.

[1] [Boundless Base public indexer and accompanying evidence archive](#)

Original collection and calculations, 10 September 2026. Boundless-Collateral-Evidence.zip preserves the snapshot and reproducible analysis. Its manifest contains every query URL and raw-response hash; final-essay/illustrative-request.json identifies Figure 3 exactly.

[2] [Multicoin Capital — Internet Labor Markets](#)

Shayon Sengupta, 10 March 2026. Prior framework on work granularity, verification and contributor economics.

[3] [Boundless — ZKC as Proving Collateral](#)

Official description of performance collateral and the typical per-request guideline; the guideline is not a measured average in this study.

[4] [Boundless — Proof Lifecycle](#)

Requestor-selected collateral, dollar-equivalent SDK inputs, unlocked fulfillment and proof batching. Inspected alongside implementation source for release semantics.

[5] [Boundless — pinned market and indexer implementation](#)

Commit 93e971a653258bb2cf1de920b103fcdfd9ef994. Market balance logic, indexer field definitions, deployment configuration and status/aggregation code were inspected. Exact additional source paths are listed in the archive README. This source pin is not a certification of deployed bytecode.

[6] [Boundless — Mining overview](#)

Staking-linked mining eligibility and reward caps, a holding channel separate from per-request bonds.

[7] [Filecoin — FIL collateral and protocol specifications](#)

Collateral types; the Miner Collaterals specification explains the sector-lifetime commitment, and Storage Mining, step 3, describes release at sector expiration. Mechanism comparison only; no Filecoin usage sample was collected.

[Miner Collaterals](#) · [Storage Mining](#)