Zero-Bond Deterrence

When Verification Is Cheap, the Correctness Bond Is Redundant

Andrew Lee
andrew@joseon.com

Abstract

Decentralized compute markets discipline a paid worker in one of two ways. Either the worker posts a slashable *bond* and a fraud proof burns it (optimistic rollups, refereed delegation), or there is no bond and the system relies on reputation and subjective trust. The first locks up capital proportional to the value at risk; the second is soft. We argue this dichotomy is not fundamental: it is an artifact of *verification cost*. A bond is what you require when catching a cheat is expensive, so you catch rarely and must punish hard when you do. Make catching cheap and the logic collapses.

We make this precise. For a rational worker who cheats only to save compute, honesty is the best response exactly when the detection margin q satisfies $q \geq \Delta/(p + \Phi)$, where Δ is the compute saved by cheating, p the fee, and Φ the franchise value of continued participation. A correctness bond b only enters this inequality as a substitute for low q: $b \geq \Delta/q - (p + \Phi)$, so it is needed *only* when verification is too sparse. When verification is cheap enough to cover (nearly) every request, $q \to 1$, and because any viable market already prices the service above the cost it saves to cheat ($\Delta < p$), the escrowed fee on the single cheated request deters on its own, at *zero bond and zero franchise*. We machine-check the result (the deterrence algebra and the martingale false-ejection bound in Z3 and Lean 4; the honesty boundary in PRISM-games) and state its scope honestly. We then observe a second consequence: once verification is the discipline, the *Sybil* cost can be the verification work itself rather than burned capital, so the anti-Sybil proof and the audit become the same computation.

1 The bond is a tax on ignorance

There is a recurring move in decentralized systems. You want to pay a stranger to do work you cannot fully check, so you make them post a bond, and you arrange that if they are ever caught cheating, the bond is destroyed. Optimistic rollups do this with sequencer bonds and fraud proofs; refereed-delegation schemes for machine learning do it with staked challengers (Arun et al., 2025); proof-of-stake networks do a softer version with slashing. The bond scales with the value at stake, and it sits idle: its whole purpose is to be larger than any profit from cheating, so that a rational worker never does.

Why a bond at all? Because catching the cheat is expensive. If you can only afford to re-check one request in a thousand, then a cheater is caught with probability 10^{-3} per offense, and to deter them you need the punishment-when-caught to be at least a thousand times the per-offense gain. That punishment cannot be the fee for one request (it is far too small), so it has to be a separately posted, much larger bond. The bond is precisely the multiplier that compensates for rarely looking.

State the principle plainly, because it generalizes past this setting. *Capital is what you stake when you cannot verify.* A correctness bond is a tax you pay for not being able to check the work; it stands in for knowledge you do not have. The interesting question is therefore not how to size the bond, but what happens to it when checking becomes cheap and you *can* look at nearly everything.

The answer, which is the subject of this paper, is that the bond's reason for existing goes away: cheap verification does not reduce the bond, it makes it redundant. What was thought to be a choice between "hard" capital security and "soft" reputation turns out to be a single axis indexed by how much it costs to check.

This is not merely an observation. Recent work has made inference verification genuinely cheap: a spot-check at a small fraction $\rho \ll 1$ of the cost of producing the output, for language models (Ong et al., 2025; Inference.net, 2025) and, in the companion to this paper, for diffusion and codec models. Cheap verification is the hypothesis; the theorem below is what it buys.

Contributions.

1. A deterrence result making precise that a correctness bond is an artifact of verification cost: honesty is the worker's dominant strategy at *zero bond and zero franchise* once cheap verification drives coverage to one, because market viability already forces the cheat-gain below the fee (§3).
2. A machine-checked treatment, formalizing the deterrence algebra and the open-ended (lifetime) false-ejection bound in Z3 and Lean 4 and reproducing the honesty boundary as a stochastic game in PRISM-games (§4), together with an honest statement of scope: the result is per-request at a *fixed* detection margin, and the regime where the margin itself is contested is a different game (§5).
3. The observation that, once verification rather than capital is the discipline, Sybil resistance can be *non-capital and dual-use*: the proof of distinct physical work that bounds an identity's share *is* the verification duty, so the anti-Sybil cost is the audit and is not burned (§6).

2 The per-request game

A worker is paid a fee p to serve a request with a model it has committed to. It chooses honest or cheat. Cheating (serving a cheaper model, a coarser quantization, or fewer sampling steps) saves it compute worth $\Delta > 0$. The payment is held in escrow and released only on an accepted result; a request found cheated forfeits the fee (the consumer is refunded by non-settlement). With probability q the request is effectively audited *and* the cheat detected; we fold coverage and detector power into one "detection margin" $q \in [0, 1]$. A worker that is caught also forfeits a *franchise* value $\Phi \geq 0$: the discounted future profit of continued participation, lost on ejection.

We assume the worker is rational and the audit is *indistinguishable*: the worker cannot tell, when serving, whether this request will be audited, so it cannot cheat only on the unaudited ones. This is the property cheap verification must deliver (audit any logged request, after the fact, on unpredictable selection); we take it as given here and treat its construction elsewhere.

Assumption 1 (Market viability)**.** The fee exceeds the compute saved by cheating: $\Delta < p$. A market in which it were not so (where the honest cost of serving exceeded the price) has no honest supply and does not clear; we are analyzing markets that exist.

3 The deterrence condition, and why the bond drops out

Expected payoff from honesty is p (serve, get paid). Expected payoff from cheating is the saved compute plus the fee when undetected, minus the forfeitures when detected:

$$U_{\text{cheat}} \;=\; \Delta \;+\; (1-q)\,p \;-\; q\,\Phi, \qquad U_{\text{honest}} \;=\; p.$$

Honesty is (weakly) preferred iff $U_{\text{honest}} \geq U_{\text{cheat}}$, i.e.

Proposition 1 (Honesty condition)**.** *At a fixed detection margin q, honest service is a worker's best response iff*

$$q \;\geq\; \frac{\Delta}{p+\Phi}.$$

Now add a correctness bond b, forfeited on detection alongside the fee and franchise. It enters in exactly one place, the punishment term, giving the condition $q \geq \Delta/(p+\Phi+b)$, equivalently

$$b \;\geq\; \frac{\Delta}{q} - (p+\Phi).$$

This is the whole role of the bond: it is the amount of *extra* punishment needed to make rare detection bite. When q is small the required b is large (the Δ/q blows up); the bond is real and necessary. But the dependence is the tell: the bond exists *only* to compensate for a small q. Drive q up and the required bond falls; at the point where $q \geq \Delta/(p+\Phi)$ the required bond is ≤ 0, meaning none is needed.

Theorem 1 (Zero-bond, zero-franchise deterrence under full coverage)**.** *Suppose verification is cheap enough to cover every request, and the detector is sound and powerful, so $q \to 1$. Then under Assumption 1 honest service is a strict best response with* no bond *($b = 0$) and* no franchise *($\Phi = 0$): forfeiture of the escrowed fee on the single cheated request already deters, because $q \cdot p \to p > \Delta$.*

Proof. Set $b = 0, \Phi = 0$ in Proposition 1: honesty holds iff $q \geq \Delta/p$. By Assumption 1, $\Delta/p < 1$. As $q \to 1$ the condition $q \geq \Delta/p$ holds strictly. □

The content is not the algebra (it is one line) but what the line says. The bond was never buying security; it was buying *coverage*, indirectly and expensively, by making the rare catch hurt enough. Buy coverage directly, by checking cheaply and often, and you need neither a bond nor even a reputation stake: the fee you were already going to pay, escrowed and forfeitable, is a sufficient punishment, because under full coverage the cheat is caught essentially every time, and one fee exceeds one cheat's saving. Capital security and reputation were two prices for the same missing thing, namely knowledge of whether the work was done, and cheap verification supplies that thing itself.

The ejection bound (why "powerful detector" is not hand-waving). A real detector accepts honest work with some false-rejection rate ε and rejects a cheat with power s. Ejecting a worker on a single rejection would eject honest workers at rate ε. Instead, accumulate a sequential probability ratio test over a worker's audited requests (Wald, 1945): under the null (honest) the likelihood ratio is a nonnegative martingale of unit initial value, so by Ville's inequality (Ville, 1939) the probability it *ever* crosses the ejection threshold $1/\beta$ is at most β, a *lifetime* false-ejection bound rather than a per-test one. Under the alternative, Wald's identity gives expected time-to-eject $\approx \log(1/\beta)/D(s\|\varepsilon)$, so a persistent cheater is removed in a handful of audits. "$q \to 1$" is shorthand for coverage near one *and* this controlled, near-zero honest ejection.

4 What is machine-checked, and what is assumed

A one-line theorem is exactly the kind of result where the danger is a hidden modeling error, not a proof gap, so we mechanized both the algebra and the part that is not algebra.

- **The deterrence inequalities** (Proposition 1, the bond expression, Theorem 1) are discharged in the Z3 SMT solver (`proofs.smt2`) as quantifier-free real-arithmetic validity: 16 checks, each annotated with its expected result, all passing. There are 15 validity `unsat`s (each claim's negation is unsatisfiable) plus one consistency `sat` witnessing that the failure mode the design rules out genuinely exists. The boundary case is re-proved in Lean 4 with mathlib (`OgongLean.lean`: `prop1_deterrence`, `cor1_zero_bond`), so the "iff" and the strictness are kernel-checked, not asserted.
- **The false-ejection bound**, the genuinely probabilistic step, is mechanized in Lean 4 (`Ogong/Ville.lean`): the supermartingale, the Ville maximal inequality, and a product-martingale lemma yield the *lifetime* bound $\leq \beta$ (`ville_bound`, `product_martingale`, `sprt_false_ejection`). The development compiles with no `sorry`, `admit`, or added `axiom`, so "honest workers are essentially never ejected" rests on a kernel-checked theorem rather than intuition about sequential tests; a Monte-Carlo harness (`verify.py`) independently reproduces the bound and the expected stopping time.
- **The honesty boundary** is also modeled as a stochastic game in PRISM-games (`ogong_audit_game.prism`, `ogong_audit_sprt.prism`), where the optimal policy reproduces the $q \geq \Delta/(p + \Phi)$ boundary. Since that boundary is already kernel-checked above, the model-checker is a corroborating cross-check, not a load-bearing step.

Every artifact named here is in the source tree and runs with the stated result; the claims are checkable, not asserted.

What remains *assumed*, not proved: Assumption 1 (an empirical property of a clearing market), audit indistinguishability (a property of the verification construction, treated in the companion work), and rationality of the worker (we model a profit-maximizer, not a griefer who burns money to disrupt; griefing is bounded by the cost of the forfeited fees, not deterred to zero).

5 Scope: a fixed margin, not an endogenous one

The honest boundary of this result is the word "fixed." Proposition 1 and Theorem 1 treat q as given. At a fixed q, the all-honest profile is the unique dominant-strategy equilibrium and there is no profitable joint deviation: a coalition of workers cannot collude to cheat, because each one's incentive is individual and the audit of one does not depend on another's behavior.

What this does *not* cover is collusion that attacks q *itself*. If the same operators that serve also verify, a coalition can try to drive the effective margin down (by passing each other's cheats), and then the fixed-q analysis no longer applies. That is a different game, the verifier-coverage game, and it needs its own bound: an operator's ability to verify its own work must be limited (by selecting verifiers from a disjoint set, and by the Sybil bound of §6), so that the achievable q stays high without anyone's cooperation. We separate the two deliberately. The zero-bond result is a statement about the per-request game once q is high; keeping q high against a strategic adversary is the coverage game, and we do not claim the former settles the latter. The deployed mechanism, accordingly, retains a small *earned* franchise ($\Phi > 0$) as belt-and-suspenders; Theorem 1 says it is not *necessary* under full coverage, not that one should run with Φ pinned to zero.

6 Sybil resistance as the verification work

Once verification rather than capital is the discipline, a second thing changes, and it is the more surprising of the two. The classic defense against an operator forging many identities to capture

reward or to self-verify is *capital*: make each identity stake, so n identities cost n stakes (Douceur, 2002). That is the same tax again: burned capital standing in for an inability to tell identities apart.

But in a compute market there is a physical quantity that is hard to forge and that we are measuring anyway: throughput. An operator's honest share of served work is bounded by the GPUs it actually has. So bound an identity's reward and self-verification probability by its share of *demonstrated physical throughput*, a proof of distinct work rather than of distinct capital, and Sybil identities buy nothing, because splitting one GPU's throughput across ten names does not create eleven GPUs' worth of work. The cost of distinctness is the cost of the work.

The dual-use is the point. The proof-of-distinct-throughput an operator must produce to hold its share *is* the verification duty it performs on others' requests: re-executing a peer's audited request is itself a unit of demonstrated work. So the anti-Sybil cost is not burned: it is the audit that makes q high in the first place. Capital-based Sybil resistance pays to throw work away; this pays for the work the system needs. (Where physical-throughput proof is unavailable or insufficient, a *non-capital* attestation of distinct hardware, such as a trusted-execution attestation binding an identity to a measured platform, serves the same role, again without locking up capital. Both are ways of paying the Sybil cost in something other than idle stake.)

This is why the network can let stake mean *priority and availability* rather than a slashable correctness bond: the correctness discipline is verification, and the Sybil discipline is work, so neither needs capital to be the thing at risk.

7 Related work

The verifier's dilemma (that a rational verifier may skip checking and just accept) is the classic obstacle (Luu et al., 2015), and the standard resolution is a forced-error or bonded-challenge scheme (TrueBit (Teutsch and Reitwießner, 2017), refereed delegation (Arun et al., 2025)). Those make the *verifier* honest with a bond; our result is about making the *worker* honest without one, by lowering the cost of the check itself rather than subsidizing a challenger. Proof-of-sampling protocols (Zhang et al., 2024) reach a Nash equilibrium for verification through randomized re-execution; they share our move of pricing detection rather than posting a correctness bond. Contemporary verified-inference systems still make the bonded choice: VeriLLM (Wang et al., 2025), a recent publicly verifiable decentralized-inference protocol, has both inferencers and verifiers post a slashable collateral stake that is forfeited on incorrect inference and locked through an unbonding delay before withdrawal. Our result is exactly that this stake is avoidable for the honest worker once the check itself is cheap. The mechanism-design backdrop is standard (Fudenberg and Tirole, 1991), and the sequential-test machinery is Wald's (Wald, 1945; Wald and Wolfowitz, 1948) with Ville's maximal inequality (Ville, 1939). Sybil resistance via cost-of-identity is Douceur (2002); our contribution is to pay that cost in verification work rather than capital.

A closing principle. If the bond is a tax on ignorance, the design rule that follows is short: do not ask participants to stake capital against being caught when you can afford to look. Spend on the looking. What you learn is worth more than what they would have staked, and, unlike the stake, it is not idle.

References

Arasu Arun, Adam St. Arnaud, Alexey Titov, Brian Wilcox, Viktor Kolobaric, Marc Brinkmann, Oguzhan Ersoy, Ben Fielding, and Joseph Bonneau. Verde: Verification via refereed delegation

for machine learning programs, 2025. Gensyn.

John R. Douceur. The Sybil attack. In *Peer-to-Peer Systems (IPTPS 2002)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

Inference.net. LOGIC: Trustless inference through log-probability verification. Technical report, `https://inference.net/blog/logic`, 2025.

Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 706–719. ACM, 2015.

Jack Min Ong, Matthew Di Ferrante, Aaron Pazdera, Ryan Garner, Sami Jaghouar, Manveer Basra, Max Ryabinin, and Johannes Hagemann. TOPLOC: A locality sensitive hashing scheme for trustless verifiable inference, 2025. Prime Intellect.

Jason Teutsch and Christian Reitwießner. A scalable verification solution for blockchains. Technical report, TrueBit, 2017.

Jean Ville. *Étude critique de la notion de collectif*. Gauthier-Villars, Paris, 1939.

Abraham Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2): 117–186, 1945.

Abraham Wald and Jacob Wolfowitz. Optimum character of the sequential probability ratio test. *Annals of Mathematical Statistics*, 19(3):326–339, 1948.

Ke Wang, Zishuo Zhao, Xinyuan Song, Zelin Li, Libin Xia, Chris Tong, Bill Shi, Wenjie Qu, Eric Yang, and Lynn Ai. VeriLLM: A lightweight framework for publicly verifiable decentralized inference, 2025.

Yue Zhang, Shouqiao Wang, Sijun Tan, Xiaoyuan Liu, Ciamac C. Moallemi, and Raluca Ada Popa. Proof of sampling: A Nash equilibrium-based verification protocol for decentralized systems, 2024. Hyperbolic.