Application-Layer Bribery Infeasibility in Order Flow Auctions: Capital Bonds as an Alternative to Multi-Proposer Consensus

Technical Note — Draft for Review

Cristian Risueño

VynX Protocol — cristian@vynx.finance

github.com/cristianrisueo/vynx-mvp

April 2026

Abstract

Fox et al. (AFT 2023) demonstrated that sealed-bid auctions on single-proposer blockchains are vulnerable to bribery attacks: priority fees (tips) reveal the private value of bids, and a monopolistic proposer can censor or reorder bid transactions to extract rents. Their proposed solution operates at the consensus layer via concurrent proposers. This document explores a complementary mechanism at the application layer implemented in VynX, a Machine-to-Machine (M2M) settlement protocol deployed on Base (Ethereum L2). The core mechanism is an Order Flow Auction (OFA) compressed to 200 milliseconds where participants (Solvers) must deposit hard capital bonds (USDC, wstETH) before competing. We argue that the combination of (i) architectural separation between off-chain auction and on-chain settlement, (ii) capital bonds as a minimum participation cost with confiscation risk (slashing), and (iii) temporal compression below the bribery coordination threshold, creates — under specific conditions — practical bribery infeasibility for both the off-chain orchestrator (Relayer) and the L2 proposer. We formalize the conditions under which this argument holds and explicitly identify open questions where consensus-level solutions remain necessary.

Keywords: Order Flow Auctions, censorship resistance, bribery infeasibility, game theory, capital bonds, sealed-bid auctions, MEV, L2, M2M settlement

1. Introduction and Motivation

The proliferation of autonomous AI agents on Ethereum Layer 2 (L2) networks has created a new category of settlement demand: machine-generated transactions, for machines, with latency and determinism requirements fundamentally different from those of human users. These transactions are executed through Order Flow Auctions (OFAs), where institutional market makers (Solvers) compete for the right to fulfill agent intents.

Fox, Pai, and Resnick [1] formally demonstrated at AFT 2023 that these auctions are structurally compromised on single-proposer blockchains. Their argument articulates two vectors: first, in an on-chain sealed-bid auction, priority fees (tips) reveal the private value of bids, de facto converting the sealed auction into an open auction with asymmetric information; second, a monopolistic proposer can be bribed to censor competing bids, guaranteeing victory for a collusive participant. Section 7.2 of their work explicitly identifies “on-chain order flow auctions and collateral liquidation auctions” as the natural extension of their censorship resistance analysis.

Their proposed solution operates at the consensus layer: replacing the single proposer with multiple concurrent proposers competing for transaction inclusion, eliminating the monopoly point where bribery is viable. This approach is theoretically robust but imposes a requirement to modify the consensus layer of the underlying blockchain — a considerable implementation constraint on existing L2s such as Base, whose centralized sequencer operates under the control of a single operator (Coinbase).

This document explores a complementary hypothesis: is it possible to construct an application-layer mechanism that achieves practical bribery infeasibility without requiring consensus modifications? We present VynX's design as a case study and formalize the conditions under which its capital bond mechanism creates such infeasibility, as well as the conditions under which the argument weakens or fails.

2. System Model

2.1. Actors

The protocol involves four classes of actors with differentiated roles and attack surfaces:

> ***Agent (A):*** *An autonomous AI system that generates cryptographically signed settlement intents (EIP-712). The agent declares a desired end state (output token, minimum amount, destination network) without specifying the execution route. Its only economic commitment is the temporary deposit of funds in the escrow contract (Origin Lock).*
>
> ***Solver (S):*** *An institutional market maker (e.g. Wintermute, GSR) that competes to execute intents. To participate in the auction, the Solver must have previously deposited hard collateral (USDC, WETH, wstETH, cbBTC, USDT) in a registry contract on Ethereum L1 (VynxRegistry.sol), subject to partial confiscation (slashing) upon breach.*
>
> ***Relayer (R):*** *The centralized off-chain orchestrator. Operates an in-memory auction engine with sub-millisecond latency. Receives intents via REST API, executes the sealed-bid auction during a 200ms window, and issues cryptographically signed settlement certificates (Vouchers) after verifying on-chain settlement. The Relayer observes all bids in plaintext during the auction window.*
>
> ***Proposer (P):*** *The L2 sequencer (Base). In the current configuration, operated centrally by Coinbase. Controls transaction inclusion and ordering in Base blocks. Block time: ~2 seconds.*

2.2. Execution Flow (Happy Path)

The lifecycle of an intent follows a strictly ordered sequence that separates the price discovery phase (off-chain) from the settlement phase (on-chain):

> ***Phase 1 — Ingestion (off-chain):*** *The Agent signs an EIP-712 intent and submits it to the Relayer via REST API. The Relayer validates the signature, verifies that tokens belong to the permitted asset whitelist, and confirms the intent exceeds the $50 USD minimum floor.*
>
> ***Phase 2 — Auction (off-chain, 200ms):*** *The Relayer broadcasts the intent to all connected Solvers via WebSocket. Solvers submit bids during the 200ms window (OFA_WINDOW). Each bid is instantly validated against the submitting Solver's Health Factor: the Solver's free collateral must be ≥ intent value × SHF_THRESHOLD (1.20). When the window expires, the Matching Engine selects the winner by offered price and SHF.*
>
> ***Phase 3 — Commitment (on-chain, 10s SLA):*** *The winning Solver has 10 seconds (SLA_COMMIT_TIMEOUT) to execute lockIntent() on VynxSettlement.sol, locking the Agent's funds*

on the origin network. This is the first point of contact with the blockchain and therefore the first attack surface exposed to the Proposer.

Phase 4 — Settlement (on-chain): *The Solver executes payment on the destination network, notifies the Relayer (PaymentNotice), and receives a signed Voucher after RPC verification. The Solver presents the Voucher to the origin contract to claim funds. The contract autonomously verifies that the Intent exists in its internal registry before releasing funds.*

2.3. Capital Bond Mechanism (Solver Health Factor)

The Solver Health Factor (SHF) is the protocol's cryptoeconomic axis. Unlike behavioral reputation systems, the SHF is a strictly financial validation evaluated in microseconds in the Relayer's RAM:

```
SHF = Free_Collateral / Intent_Value
```

Where Free_Collateral = Total_Collateral − In_Flight_Capital. A bid is admitted if and only if SHF ≥ SHF_THRESHOLD (1.20). This imposes a structural 20% overcollateralization as a minimum participation cost. Additionally, MAX_SOLVER_EXPOSURE (80%) prevents a Solver from committing more than 80% of its total bond to simultaneous intents, guaranteeing a 20% free reserve.

Permitted collateral is restricted to low-volatility assets (USDC, USDT, WETH, cbBTC, wstETH) deposited on Ethereum L1 through Restaking protocols (EigenLayer, Symbiotic). Upon Deadline breach (15 minutes), an automatic confiscation of 10% of the intent's nominal value is executed from the Solver's bond.

3. Threat Model

We adopt the threat model of Fox et al. [1] and extend it to cover the off-chain attack surface introduced by the Relayer. We define three bribery vectors that a rational adversary could attempt to exploit:

3.1. Vector R: Relayer Bribery

The Relayer is the actor with maximum information: it observes all bids in plaintext during the 200ms window. An adversary (collusive Solver S_c) could attempt to bribe the Relayer to obtain one of the following advantages:

R.1 — Bid leakage: *S_c pays the Relayer to receive competitors' bids before the window closes, allowing S_c to adjust its offer to win by the minimum margin and capture the surplus.*

R.2 — Selective exclusion: *S_c pays the Relayer to discard bids from specific competitors during matching, reducing effective competition.*

R.3 — Voucher fabrication: *S_c pays the Relayer to issue a Voucher without S_c having executed the actual settlement on the destination network, enabling fraudulent withdrawal of escrowed funds.*

3.2. Vector P: Proposer Bribery

The Base Proposer (Coinbase sequencer) controls transaction inclusion. Following the analysis of Fox et al. [1], an adversary could attempt:

P.1 — lockIntent() censorship: *Prevent a legitimate winning Solver from executing the fund lock within the 10-second SLA window, forcing a re-auction where the collusive Solver has an advantage.*

P.2 — Settlement front-running: *Reorder claimFunds() transactions to extract value from the settlement sequence.*

3.3. Vector C: Relayer-Proposer Collusion

> ***C.1 — Coordinated attack:*** *The Relayer leaks the winning bid to the Proposer, who selectively censors the legitimate winner's lockIntent() transaction. The Relayer then re-executes the auction or declares a new collusive winner. This is the most sophisticated attack vector and combines the informational advantages of both actors.*

4. Bribery Infeasibility Argument

We analyze each threat vector and argue that VynX's design creates conditions under which bribery is economically irrational (practical infeasibility) or technically ineffective, without requiring modifications to the underlying L2 consensus.

4.1. Against Vector R: Relayer Bribery Infeasibility

4.1.1. Against R.1 (Bid Leakage)

In a classical sealed-bid auction, knowing competitors' bids allows the colluder to bid just above the second-highest offer, capturing the entire surplus. However, in VynX winner selection does not depend exclusively on the offered price: the Matching Engine jointly weighs the OutputAmount and the Solver Health Factor.

This implies that, even with perfect information about competing bids, S_c cannot win the auction without satisfying the SHF ≥ 1.20 requirement and having sufficient free collateral. Leaked information reduces the captured surplus but does not eliminate the participation cost (the capital bond remains locked and exposed to slashing). The expected value of bribery reduces to the marginal difference between the optimal bid with perfect information and the optimal bid without it, net of the bribery cost to the Relayer and the opportunity cost of locked capital.

Additionally, the 200ms window imposes a critical temporal constraint: the bribery communication channel (Relayer $\rightarrow S_c$) and the adjusted bid generation must complete within the same window during which competitors' bids are still being received. Unlike on-chain auctions where the bidding period spans multiple blocks (30+ seconds in protocols like CoW), compression to 200ms reduces the available time for bribery coordination to a level where the risk of coordination failure is significant.

4.1.2. Against R.2 (Selective Exclusion)

Selective exclusion of legitimate Solvers reduces competition and therefore execution quality for Agents. However, this attack has a direct cost to the Relayer: its per-transaction revenue is the Take Rate (10 bps on InputAmount), which is independent of which Solver wins. The Relayer captures no additional surplus by favoring a specific Solver. Its economic incentive is to maximize total transaction volume, which requires maintaining competitive execution prices that attract more Agents. Systematic exclusion degrades execution quality, reduces volume, and therefore the Relayer's own revenue.

More formally: let V_R be the Relayer's per-transaction revenue (Take Rate × InputAmount), B the bribe offered by S_c, and ΔV the future volume reduction caused by price degradation. Exclusion is rational for the Relayer only if $B > \Sigma(\Delta V_t)$ for all future periods t, where ΔV_t is the revenue loss in period t derived from reputational damage. In a market with multiple order flow sources (AgentKit, proprietary SDKs), reputational loss has a compounding effect that substantially raises the bribery cost.

4.1.3. Against R.3 (Voucher Fabrication)

This is the most critical vector and the only one with potential for direct capital loss to Agents. VynX mitigates it through a deterministic self-defense mechanism at the smart contract level (On-chain Determinism).

The VynxSettlement.sol contract does not blindly trust the Relayer's signature. Upon receiving a claimFunds(Voucher) call, the contract autonomously verifies three conditions: (a) that the IntentID referenced by the Voucher exists in its internal deposit registry (Origin Lock), (b) that the Solver address in the Voucher matches the deposit record, and (c) that the Intent's state is LOCKED. If any condition fails, the transaction reverts atomically and emits the SuspiciousRelayerActivity event.

This cross-verification turns Voucher fabrication into an attack that requires, as a precondition, the existence of a legitimate prior deposit by the Agent. A compromised Relayer can, at most, issue a valid Voucher for an intent that was genuinely deposited but whose destination settlement was not correctly verified. The attack surface reduces from "arbitrary fabrication" to "negligent settlement verification" — a vector quantitatively bounded by the number of active intents at any given moment.

4.2. Against Vector P: Proposer Bribery Mitigation

The analysis by Fox et al. [1] demonstrates that, in on-chain auctions, the Proposer can extract the private value of bids from priority fees (tips), since higher bids correlate with higher tips to ensure inclusion. VynX neutralizes this vector through a fundamental design principle: the auction never touches the blockchain.

4.2.1. Architectural Separation

In VynX, the price discovery phase (Phase 2: the 200ms auction) occurs entirely off-chain in the Relayer's RAM. Bids are transmitted via encrypted WebSocket and are never recorded in the mempool or any L2 block. The Base Proposer has no visibility into individual bids, their amounts, or the identity of non-winning participants.

The only transactions the Proposer observes are lockIntent() (Phase 3) and claimFunds() (Phase 4). These transactions reveal the auction outcome (who won and for how much), but not the losing bids. The information the Proposer can extract is therefore equivalent to that of a public observer after the close of any auction. The tip $\rightarrow$ bid_value leakage channel identified by Fox et al. is eliminated because there are no tips associated with individual bids: the only tip is the winning Solver's when executing lockIntent(), which reveals only its own offer.

4.2.2. lockIntent() Censorship

The censorship vector remains theoretically viable: the Proposer could refuse to include the winning Solver's lockIntent() transaction. However, VynX incorporates temporal margins that mitigate this attack:

The SLA window is 10 seconds (SLA_COMMIT_TIMEOUT). With a block time of ~2 seconds on Base, this grants the winning Solver approximately 5 consecutive inclusion opportunities. For censorship to succeed, the Proposer must censor the transaction across 5 consecutive blocks. The cost of censorship scales linearly with the number of blocks: each censored block implies forgoing the legitimate transaction's fees and risking detection of the censorship pattern by external observers.

Additionally, if censorship succeeds and the SLA expires, the winning Solver receives an operational penalty (Jail Time Level 1) but the Agent loses no capital: the intent returns to FAILED state and the Agent retains its funds intact (the Origin Lock was never executed). Censorship damage is limited to an operational delay and an unjust penalty on the legitimate Solver, not a capital loss.

4.2.3. claimFunds() Front-running

VynxSettlement.sol implements strict verification in the claim function: only the Solver whose address matches the one recorded in the Voucher can execute the withdrawal. A Proposer observing the claimFunds() transaction

in the mempool cannot front-run it with their own address because the contract will verify the cryptographic identity of the beneficiary against the Relayer-signed Voucher. Settlement front-running is technically futile.

4.3. Against Vector C: Relayer-Proposer Collusion

The coordinated attack (C.1) is the most sophisticated vector and the one presenting the most honest limitations of our argument. In theory, a collusive Relayer could communicate to the Proposer the winning Solver's identity and the expected lockIntent() transaction, enabling targeted censorship.

However, we argue that the practical infeasibility of this attack holds under the following compound condition:

> ***Collusion Infeasibility Condition (CIC):*** *The coordinated attack is economically irrational if the total cost of collusion (Relayer bribe + Proposer bribe for N blocks + detection risk + Relayer reputational loss) exceeds the capturable surplus by S_c in a single auction, weighted by the probability of successful censorship sustained for ≥ 5 consecutive blocks.*

The key to this argument is that per-intent surplus is bounded by the spread between the best and second-best bids. In a competitive market with multiple institutional Solvers, this spread tends to compress. Meanwhile, the collusion cost includes a fixed component (coordination between two actors) and a variable component (censorship across multiple blocks) that does not scale favorably with the number of attempts.

We explicitly acknowledge that this argument is one of economic infeasibility, not technical impossibility. A sufficiently capitalized adversary with privileged access to both the Relayer and the Proposer could, in theory, execute this attack for high-value intents. Quantifying the exact intent-value threshold above which collusion becomes rational is an open question dependent on empirical market parameters.

5. Validity Conditions and Argument Limits

The infeasibility argument presented in Section 4 holds under a specific set of conditions. Intellectual honesty requires making explicit where the application-layer mechanism is sufficient and where it is not.

5.1. Conditions Under Which the Argument Holds

> ***Competitive Solver market:*** *The surplus compression argument requires ≥ 3 institutional Solvers actively competing on a regular basis. In a market with a single Solver, bid leakage has no value (there are no competitors to leak from) but exclusion and collusion become trivial.*
>
> ***Relayer as a reputational business:*** *Relayer bribery infeasibility depends on its business model (Take Rate) generating recurring revenue that exceeds the value of any one-time bribe. This requires sufficient transaction volume for cumulative revenue to dominate.*
>
> ***SLA window > N × BlockTime:*** *Proposer censorship resistance depends on the SLA window (10s) offering multiple inclusion opportunities. If Base were to change to 5-second blocks, the margin would shrink to 2 opportunities, significantly weakening the guarantee.*
>
> ***Collateral immobilized during auction:*** *SHF_THRESHOLD must be evaluated against collateral effectively locked on L1, not future commitments. On-chain verification by VynxRegistry.sol guarantees this condition.*

5.2. Conditions Under Which the Argument Weakens or Fails

Cold Start (< 3 active Solvers): *During the protocol's initial phase, the number of active Solvers is likely insufficient to generate real competition. In this scenario, Relayer bribery infeasibility weakens substantially: with a single Solver, the auction degenerates into unilateral acceptance and the Relayer loses reputational incentive to maintain neutrality.*

Extreme-value intents: *For intents whose value significantly exceeds the combined collusion cost (Relayer bribe + Proposer bribe for 5 blocks), attack C.1 becomes economically rational. This threshold depends on empirical coordination costs and is an unknown requiring mainnet calibration.*

Persistent Proposer centralization: *The censorship resistance argument assumes that censoring ≥ 5 consecutive blocks has a significant cost. On an L2 with a centralized sequencer (current Base), this cost is fundamentally reputational, not cryptoeconomic. If the sequencer operator (Coinbase) had incentives aligned with a specific Solver, the guarantee collapses. This is the most honest limitation of the application-layer approach: it cannot substitute censorship guarantees that require consensus decentralization.*

Relayer as single point of informational failure: *Unlike consensus-level solutions where bids are cryptographically encrypted (e.g. commit-reveal with TEEs), VynX's Relayer observes all bids in plaintext. The architectural separation protects against the Proposer but not against an internally compromised Relayer. A future extension could explore using TEEs (Trusted Execution Environments) to run the Matching Engine, eliminating the need to trust the Relayer operator.*

6. Open Questions

This document raises hypotheses that require formal and empirical validation. We identify the following questions as the most critical for determining the viability of the application-layer approach:

Q1. Does temporal compression substitute for cryptographic bid privacy?

Fox et al. propose consensus-level mechanisms to guarantee bid privacy. VynX argues that a 200ms window compresses the time available to coordinate bribery to the point of impracticality. Is there a formal temporal compression threshold below which bribery coordination becomes NP-hard or computationally intractable? Or does speed only raise the cost without eliminating the possibility?

Q2. What is the intent-value threshold above which Relayer-Proposer collusion becomes rational?

The Collusion Infeasibility Condition (CIC) defined in Section 4.3 depends on empirical parameters (coordination cost, reputational cost, competitive spread). Is it possible to derive a closed-form bound for this threshold as a function of the number of active Solvers, average volume, and L2 gas costs?

Q3. Can Restaking Vault Adapters introduce slashing escape vectors?

VynX delegates collateral custody to external Restaking protocols (EigenLayer, Symbiotic) through abstracted interfaces (Vault Adapters). Are there vectors where a Solver could exploit differences in these protocols' withdrawal mechanisms to evade confiscation within the REBALANCE_EPOCH window (7 days)?

Q4. What formal guarantees does a TEE in the Matching Engine offer versus the current trusted-Relayer model?

If the Matching Engine were executed inside a TEE (e.g. Intel SGX, ARM TrustZone), bids would remain encrypted even to the Relayer operator. Would this architectural change elevate the guarantee from practical infeasibility to technical impossibility? Or do side-channel attacks on TEEs

invalidate this promise for 200ms windows?

Q5. Does the Proposer censorship vector survive progressive sequencer decentralization on Base?

If Base transitions to a decentralized sequencer model (as proposed in its roadmap), does the application-layer censorship resistance guarantee become redundant with the consensus-layer guarantee, or do both layers provide complementary protection against different adversary classes?

7. Reference Implementation

The mechanisms described in this document are implemented in a functional MVP deployed on Base Sepolia (testnet). The source code is public and verifiable:

VynxSettlement.sol contract: *Deployed and verified on Base Sepolia. Implements Origin Lock, deterministic self-defense (Section 4.1.3), Take Rate deduction, and alert event emission. Address: 0xA0d462b84C2431463bDACDC2C5bc3172FC927B0B*

Relayer Engine (Go): *Modular monolith with zero-DB architecture on the Hot Path. Implements the Gatekeeper, 200ms Matching Engine, and Witness Service for settlement verification.*

AgentKit Plugin (@vynx/agentkit-plugin): *Native integration with Coinbase AgentKit and CDP MPC Wallets. Enables AI agents to generate signed EIP-712 intents and submit them to the Relayer in a single call.*

End-to-End Demo: *The* `make reviewer-demo` *command spins up the entire stack via Docker and executes the complete cycle: MPC wallet provisioning → EIP-712 signing → 200ms auction → voucher emission → on-chain settlement.*

`Repository: github.com/cristianrisueo/vynx-mvp`

8. Conclusion

We have presented a practical bribery infeasibility argument for application-layer order flow auctions, complementary to the consensus-level solutions proposed by Fox et al. [1]. The capital bond mechanism implemented in VynX creates three economic barriers against bribery: minimum participation cost (SHF ≥ 1.20), confiscation risk (10% slashing), and temporal compression (200ms window).

We have been explicit in distinguishing between practical infeasibility (the attack cost exceeds the expected benefit under reasonable market conditions) and technical impossibility (the attack is cryptographically unrealizable). Our mechanism achieves the former, not the latter. Consensus-level solutions remain necessary for stronger guarantees, particularly in scenarios involving high-value intents and persistent sequencer centralization.

The specific contribution of this work is demonstrating that, for the class of medium-value M2M transactions (the dominant use case for AI agents in the near term), an application-layer mechanism can provide practically sufficient bribery resistance guarantees without imposing consensus modification requirements on the underlying L2. This opens an immediate implementation pathway on existing L2s while consensus-level solutions mature.

References

[1] Fox, E., Pai, M.M., Resnick, M. (2023). “Censorship Resistance in On-Chain Auctions.” 5th Conference on Advances in Financial Technologies (AFT 2023). LIPIcs, Vol. 282. doi: 10.4230/LIPIcs.AFT.2023.20

[2] Fox, E., Robinson, D. (2024). “Putting All Your Calls in One Basket: Orderflow Quality as a Central Design Problem.” Variant Fund Research.

[3] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., Juels, A. (2020). “Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability.” IEEE Symposium on Security and Privacy.

[4] EigenLayer. (2023). “EigenLayer: The Restaking Collective.” Whitepaper. eigenlayer.xyz

[5] Coinbase. (2024). “AgentKit: Every AI Agent Deserves a Wallet.” github.com/coinbase/agentkit

[6] VynX Protocol. (2026). “P1–P5: Protocol Technical Documentation.” Internal documentation.

Contact: cristian@vynx.finance | Code: github.com/cristianrisueo/vynx-mvp