# A High-Performance Cross-Chain Protocol for Perpetual Markets

**Abstract.** This paper introduces a novel architecture for fully decentralized perpetual swaps that bridge the performance gap with centralized venues while leveraging Bitcoin's security for final settlement. We present a hybrid execution layer combining Solana's high-throughput virtual machine with Bitcoin settlement, achieving sub-second cross-venue routing computations. Our decentralized sequencer network provides censorship-resistant transaction ordering, while finality is secured via Bitcoin anchoring through established bridging mechanisms. We formulate an adapted Almgren-Chriss execution model optimized for cryptocurrency derivative microstructure and introduce an AI-assisted interface using deep reinforcement learning for trade optimization. The system incorporates robust risk management for leverage up to 150x, balancing capital efficiency with systemic safety. Finally, we establish principles for natural language interfaces to enhance usability in complex financial domains. Our modular design demonstrates a practical path toward resolving the trade-offs between execution speed, settlement security, and true decentralization in global derivatives trading.

## 1. Introduction

Market microstructure is the academic field concerned with the economic forces that influence trades, quotes, and prices, and the processes by which transaction prices converge to, or deviate from, long-term equilibrium values due to market frictions.[1] Within the cryptocurrency domain, no instrument is more central to this study than the perpetual swap. Perpetual swaps, or "perps," are derivative contracts that allow traders to speculate on asset prices without an expiration date, and they have come to dominate the digital asset landscape, accounting for approximately 93% of all crypto derivatives trading volume.[2] Their trading volumes frequently surpass those of the underlying spot

markets, establishing them as primary venues for price discovery and liquidity concentration.[3] The defining feature of these instruments is the funding rate mechanism, a periodic payment exchanged between long and short positions, typically every eight hours, which serves to tether the contract's price to the underlying spot price.[2] This mechanism is a powerful driver of market behavior, inducing a characteristic U-shaped pattern in both trading activity and bid-ask spreads within each funding cycle as traders position themselves around the payment event.[2] The profound influence of perpetual contracts extends beyond their own markets; empirical evidence suggests that their introduction actively shapes the microstructure of the underlying spot markets. The availability of high leverage, the ease of short selling, and continuous 24/7 trading create a highly efficient environment for informed traders, which in turn boosts spot market liquidity and price efficiency.[2] However, this efficiency comes at a cost: the very mechanisms that attract sophisticated participants also lead to increased trading costs and heightened adverse selection for uninformed traders.[2] This fundamental trade-off between market efficiency and transaction cost is a central challenge that any advanced trading protocol must address. A protocol that can significantly reduce the frictions associated with perpetuals trading will not merely capture existing market share but will likely amplify these microstructural effects, potentially becoming the dominant global venue for price discovery.[3]

## 1.1 DEXs

The design of decentralized exchanges is constrained by a fundamental challenge that can be termed the "Execution Trilemma": the difficulty of simultaneously optimizing for centralized exchange level performance, characterized by low latency and high throughput; true decentralization, ensuring censorship resistance and global market coverage; and ultimate settlement security on a globally recognized, robust ledger. Existing solutions invariably compromise on at least one of these dimensions.

High-performance DEXs, particularly those built as application-specific blockchains or "app-chains," often achieve their speed by employing a centralized or permissioned sequencer.[6] This single entity is responsible for ordering transactions, creating a single point of failure and a vector for censorship or malicious reordering (MEV), thereby sacrificing decentralization for performance.[8] Conversely, protocols that prioritize decentralization by settling directly on a base layer like Ethereum often inherit its performance limitations, such as high latency and variable transaction fees. These limitations can render certain trading strategies, especially those involving high frequency or high leverage, economically unviable and competitively disadvantaged compared to their CEX counterparts. This architectural tension has left a significant gap in the market for a protocol that can deliver the performance of a CEX without compromising on the core tenets of decentralization and security.

### 1.2 Modularity

This paper posits that the execution trilemma can be resolved through a modular architecture that separates the core functions of a trading system, allowing each component to be optimized for its specific purpose without compromise.

The proposed protocol comprises three distinct layers:

1. *Execution Layer*: A custom Layer 2 built using the Solana Virtual Machine, engineered for parallel transaction processing and sub-75ms computational latency, providing the raw performance necessary for a high-frequency, low-latency trading environment.
2. *Sequencing Layer*: A decentralized, permissionless network of nodes responsible for fairly ordering user transactions. This layer provides censorship resistance and mitigates the risks associated with centralized sequencers.
3. *Settlement Layer*: The Bitcoin blockchain, utilized as the ultimate arbiter of state and finality. By employing an optimistic rollup design, the protocol anchors its security in the most robust and decentralized ledger in existence.

This separation of concerns allows the protocol to achieve CEX-level execution speed on its dedicated L2, maintain censorship resistance through its decentralized sequencing layer, and inherit the unparalleled security of Bitcoin for final settlement. The subsequent sections will provide the detailed mathematical, architectural, and security foundations to substantiate this thesis.

## 2. Mathematical Foundations

The foundational model for optimal trade execution is the framework developed by Almgren and Chriss, which seeks to find a trading trajectory that minimizes a linear combination of execution costs and risk.[9]

The standard Almgren-Chriss model assumes the unaffected security price Sk follows a discrete arithmetic random walk, with a permanent market impact term that is a function of the trading rate:

$$S_k = S_{k-1} + \sigma\sqrt{\tau}\xi_k - \tau g(v_k)$$

To apply this framework to the unique microstructure of cryptocurrency perpetual swaps, this model must be significantly extended. The high volatility, distinct fee structures, and unique carrying costs of perpetuals necessitate a more nuanced formulation.

The adapted model must incorporate several crypto-specific factors:

- Exchange Fees: A term, $\phi(n_k)$, representing the taker or maker fees, which are non-trivial and can vary based on trading volume and order type.[11]
- Bid-Ask Spread: A cost component, $\epsilon_k$, representing the price paid for liquidity by crossing the bid-ask spread, a dominant factor in execution costs on many exchanges.[11]
- Funding Rates: A stochastic process, $f_k$, modeling the periodic funding payments that act as a significant carrying cost or revenue stream for the duration of the position.[2]
- High Volatility and Non-Static Drift: The model must accommodate the characteristically high volatility ($\sigma$) and the presence of strong, time-varying drift ($\mu_k$) common in digital assets, which a static model fails to capture effectively.[11]

## 2.1 Funding Rates

The linear impact functions often assumed in basic models are an oversimplification. Empirical evidence suggests that market impact, particularly in less liquid markets, is better described by a non-linear function, such as a power law of the trading rate, $h(v_k)=c|v_k|^{\delta}$, where $h(\cdot)$ is the temporary impact function.[9]

Furthermore, in a high-frequency environment, latency is not merely a technical inconvenience but a quantifiable risk factor. The state of the limit order book (LOB) can change dramatically within milliseconds. Therefore, we introduce a latency penalty function, $\Lambda(\Delta t)$, which increases the expected execution cost based on the time delay Δt between observing a market state and executing a trade. This function mathematically formalizes the critical need for the sub-75ms computational performance of the protocol's execution layer.

The complete implementation shortfall minimization problem is then formulated as the objective to minimize the expectation and variance of the total execution cost. The goal is to find the optimal trade list $\mathbf{n}=\{n_1,\ldots,n_N\}$ that minimizes:

$$\min_{\mathbf{n}} \left( \mathbb{E}\left + \lambda \text{Var}\left \right)$$

where λ is the trader's risk aversion parameter.

The realized execution price, S~k, for a sell order is modeled as:

$$S_k = S_{k-1} + \sigma\sqrt{\tau}\xi_k - \tau g(v_k)$$

The dynamics of the unaffected price $S_k$ are given by:
$$ S_k = S_{k-1} + \mu_k\tau + \sigma_k\sqrt{\tau}\xi_k - \tau g(v_k) - f_k \cdot \mathbb{I}(\text{funding event}) $$ where $I(\cdot)$ is an indicator function for funding payment events. This comprehensive model provides a more realistic foundation for deriving optimal execution strategies in the crypto derivatives market.

## 2.2 Optimal Execution

The static optimization of a pre-determined trading trajectory, as in the original Almgren-Chriss model, is ill-suited for the highly volatile and reflexive nature of cryptocurrency markets.[11] A superior approach is to frame the problem dynamically, allowing the trading strategy to adapt to changing market conditions in real-time. This is achieved by modeling the optimal execution problem as a Markov Decision Process.[12] This reformulation represents a paradigm shift from calculating a fixed execution path to learning a dynamic, state-aware policy, which is a prerequisite for a truly AI-native system.

The MDP is formally defined by the tuple $\langle S,A,P,R,\gamma \rangle$:

- State Space ($S$): The state $s_t \in S$ is a vector that captures all relevant information at time t. It includes the remaining inventory to be executed, the time remaining until the deadline, a representation of the current limit order book state, recent realized volatility, and the predicted next funding rate.
- Action Space ($A$): The action $a_t \in A$ is the decision made by the agent in state $s_t$. It consists of the quantity of contracts to trade in the next time step and the distribution of this quantity across different order types and venues.[15]
- Transition Probability ($P$): The function $P(s_{t+1}|s_t,a_t)$ defines the probability of transitioning to state $s_{t+1}$ after taking action at in state $s_t$. This is governed by the stochastic price and LOB dynamics defined in the adapted model from Section 2.2.
- Reward Function ($R$): The reward $r_t$ received after taking action $a_t$ is defined as the negative implementation shortfall over that single step. It captures the immediate execution cost, penalizing adverse price movements, market impact, fees, and spread crossing.
- Discount Factor ($\gamma$): A factor $\gamma \in$, typically close to 1, which prioritizes immediate rewards while still accounting for the total long-term execution cost.

The solution to this MDP is an optimal policy, $\pi^*: S \rightarrow A$, which maps any given market state to the action that maximizes the expected cumulative reward. This policy provides a dynamic, adaptive trading strategy that can react intelligently to real-time market conditions, a significant improvement over static trajectories. This MDP formulation serves as the theoretical foundation for the AI-assisted trading agent detailed in Section 5.

## 3. Protocol

The architecture is a deliberate response to the execution trilemma, separating concerns into three specialized layers to achieve performance, true decentralization, and security without compromise. The high-leverage nature of the platform places extreme demands on both performance and security; a millisecond delay in liquidation processing can be as catastrophic as a settlement layer re-organization. This architecture synergistically combines the strengths of different technologies to meet these demands.

### 3.1 Execution Layer

The execution layer is a custom, application-specific Layer 2 designed for maximum performance, built using the Solana Virtual Machine (SVM). The choice of SVM is predicated on its Sealevel runtime, a key innovation that enables parallel processing of transactions.[17] Unlike single-threaded virtual machines like the EVM, where transactions are processed sequentially, Sealevel can execute multiple non-conflicting transactions concurrently across all available CPU cores of a validator node.[20]

This parallelism is achieved because Solana transactions are required to declare upfront all the accounts they intend to read from or write to during execution.[21] With this information, the scheduler can construct a dependency graph and execute all transactions with non-overlapping state access requirements in parallel. This architecture is exceptionally well-suited for a derivatives exchange, where a large volume of operations, such as trades on different currency pairs, oracle price updates, funding rate calculations, and liquidations of independent positions, can be processed simultaneously, dramatically increasing throughput and reducing block processing latency.[17]

Public Solana network benchmarks demonstrate block times of approximately 75ms and sustained, real-world throughput of 102,000-104,000 transactions per second (TPS) under heavy load.[24] As a dedicated L2, our execution layer can be further optimized, stripping away unnecessary general-purpose functionality to achieve the target of sub-50ms latency for complex internal computations like cross-venue order routing and multi-position risk updates. This layer serves as the "fast path" for all time-sensitive operations, while transaction data is batched for subsequent ordering and settlement.

### 3.2 Sequencer

To counteract the centralization and censorship risks inherent in many L2 designs that rely on a single sequencer [6], our protocol incorporates a dedicated, decentralized sequencing layer. This layer's sole responsibility is to receive transactions from users, agree on a canonical ordering, and assemble these transactions into batches that are then passed to the settlement layer.

The architecture of this layer is inspired by emerging shared sequencer networks such as Astria and Espresso.[8] It consists of a permissionless set of sequencer nodes that anyone can join by staking a protocol-native token. These nodes operate a Byzantine Fault Tolerant consensus protocol, such as CometBFT, to agree on the order and content of each batch.[8] The right to propose a new batch rotates among the staked nodes, preventing any single entity from having a persistent monopoly on transaction inclusion and thus ensuring strong censorship resistance and liveness for the system.[8] This layer is also the critical point for implementing front-running mitigation, as it will enforce a fair ordering policy on the encrypted transaction intents it receives, a process detailed in Section 4.1.

### 3.3 Settlement

The final and most critical layer provides settlement security by anchoring the protocol's state to the Bitcoin blockchain. The choice of Bitcoin is motivated by its unmatched security budget, decentralization, and historical liveness, making it the ideal foundation for a system intended to secure substantial value.[30] The mechanism for this connection is an optimistic rollup.[6] Transaction data from the execution layer is batched by the sequencing layer and posted to the Bitcoin blockchain. The state transitions computed on the L2 are assumed to be correct by default, the "optimistic" assumption, which allows for high throughput without waiting for direct on-chain verification of every computation.[34]

The core innovation of this layer is the method for enforcing the validity of these state transitions. We propose using a framework inspired by BitVM to enable the verification of fraud proofs on Bitcoin without requiring a consensus-breaking hard fork.[36]

This system operates on a prover-verifier model:

- *Provers:* A set of L2 operators are responsible for executing transactions and posting the new state root to Bitcoin, along with a bond. In doing so, they make a verifiable claim that the state transition was valid.
- *Verifiers:* Any party running a full node of our L2 can act as a verifier. If a verifier detects an invalid state transition, they can initiate a challenge.
- *Challenge-Response Protocol*: The challenge triggers an interactive game on the Bitcoin blockchain, constructed using a pre-signed set of transactions and a large Taproot tree. This game allows the verifier to force the prover to reveal the specific, atomic computational step that was executed incorrectly.[36]
- *Punishment*: If the prover fails to correctly respond to the challenge or is proven to have equivocated, their bond is slashed on the Bitcoin L1. This on-chain, economic punishment serves as the fraud proof, allowing the L2 network to disregard the invalid block and revert to the previous valid state.

To overcome the two-party limitation of the original BitVM proposal, this architecture incorporates concepts from BitVM 2, which enables permissionless verification.[39] This enhancement means that any observer, not just a pre-defined set of verifiers, can initiate a challenge. This open verification model drastically improves the protocol's security and decentralization, as an attacker would need to be confident that no single honest party in the world would detect their fraud.

## 4. Security

To provide execution quality that rivals centralized exchanges, the protocol is designed not just as an isolated liquidity pool but as a sophisticated aggregation engine. The optimal execution algorithm, derived from the mathematical framework in Section 2, will intelligently route user orders across both the protocol's native order book and other compatible on-chain liquidity venues. This ensures that trades are filled at the best possible price by tapping into a wider pool of liquidity.

A primary challenge in on-chain trading is the prevalence of Miner Extractable Value (MEV), particularly front-running and sandwich attacks, where malicious actors exploit their view of pending transactions to profit at the expense of users.[40] Our protocol implements a robust, two-tiered defense against such manipulation.

First, users submit their trading intentions as encrypted payloads. This practice, known as an encrypted mempool, prevents MEV searchers and potentially malicious sequencers from inspecting the content of transactions, such as the asset, size, direction, and slippage tolerance, before they are ordered.[43] This information asymmetry is the root cause of most MEV, and encryption effectively eliminates it at the source.

Second, to prevent ordering manipulation based on other metadata, the protocol enforces fair ordering through a "commit-reveal" scheme. The decentralized sequencers must first come to a consensus on the order of a block of encrypted transactions. Only after this order has been finalized and immutably committed does a distributed committee of nodes collaboratively generate the decryption key. This is achieved using a threshold cryptosystem, where the private key is sharded among the committee members, and a threshold number of them must cooperate to reconstruct it.[40] This cryptographic dependency ensures that no single party can both see the content of transactions and determine their order, making intelligent front-running impossible.[47] To further harden the system against high-frequency attacks, a Verifiable Delay Function (VDF) is employed to introduce a provable, un-parallelizable time delay between the production of consecutive blocks, ensuring a minimum time for information propagation and leveling the playing field for all participants.[48]

## 4.1 Oracle

The provision of 150x leverage imposes extreme requirements on the protocol's oracle system. In such an environment, minor price fluctuations can trigger liquidations, making the accuracy, frequency, and latency of price updates a matter of systemic solvency. Traditional oracle solutions, which update on-chain prices on a slow, periodic basis, are fundamentally inadequate and would introduce unacceptable risk.

Our protocol's oracle architecture is designed for this high-stakes environment. It will be a hybrid system that leverages the robust, decentralized data aggregation of established oracle networks like Chainlink to source high-quality price feeds from a wide array of off-chain exchanges and data providers.[30] These feeds will be ingested directly by nodes on the high-performance SVM L2. On this execution layer, the price data can be updated, cross-referenced, and validated at sub-second intervals.

The L2 itself will perform a final, on-chain aggregation and sanity check before the price is used to mark user positions to market and trigger liquidations. This design minimizes the critical latency between a price change occurring in the broader market and that change being acted upon by the protocol's risk engine.

## 4.2 Risk Management

Maintaining protocol solvency with leverage as high as 150x requires a mathematically rigorous and conservative risk management framework. This framework encompasses the measurement of risk, the dynamic calculation of margin requirements, and the continuous stress testing of the system's backstops.

The protocol will eschew the widely used but flawed Value at Risk (VaR) metric. VaR merely provides a loss threshold for a given probability but fails to quantify the magnitude of losses that exceed this threshold. Furthermore, it is not a coherent risk measure, meaning it can discourage diversification, and its optimization is often an ill-posed problem for portfolios containing non-linear instruments like derivatives.[51] Instead, the core risk metric will be Conditional Value at Risk (CVaR), also known as Expected Shortfall. CVaR measures the expected loss in the tail of the distribution—specifically, the average loss given that the loss has already exceeded the VaR threshold. CVaR is a coherent risk measure and is far better suited to capturing the extreme tail risk characteristic of volatile cryptocurrency markets.[51]

Margin requirements for user positions will not be static. They will be calculated dynamically based on the CVaR of each user's entire portfolio. This calculation will use advanced statistical models that account for the empirically observed properties of crypto asset returns, such as heavy tails and

asymmetry, by employing distributions like the stable Paretian distribution rather than assuming normality.[56]

Finally, the protocol's insurance fund, the ultimate backstop against losses from liquidations that execute below the bankrupt price, will be continuously evaluated through a rigorous program of stress testing and scenario analysis.[58] A series of severe but plausible market scenarios will be simulated, including multi-standard-deviation price shocks, the de-pegging of major stablecoins, and periods of extreme, sustained volatility. These simulations will assess the impact on the system's solvency and ensure that the insurance fund is adequately capitalized to withstand such events, thereby protecting solvent users from socialized losses.[61]

## 5. Human-Computer Interaction Layer

The protocol's AI-native design is realized through an AI assistant that helps users formulate and execute optimal trading strategies. This system is a direct implementation of the solution to the Markov Decision Process (MDP) formulated in Section 2.3. The core methodology employed is Deep Reinforcement Learning, a technique that excels at solving complex, sequential decision-making problems in dynamic environments without requiring a complete, analytical model of the underlying market dynamics.[63]

A DRL agent will be trained in a high-fidelity simulation environment built from historical and real-time limit order book data. The agent's objective is to learn an optimal policy, $\pi^*$, that maps market states to trading actions to maximize the expected cumulative reward, which is defined as the negative of the total implementation shortfall. State-of-the-art policy gradient algorithms, such as Proximal Policy Optimization  or Deep Deterministic Policy Gradient, will be used for training, as they have demonstrated strong performance in similar financial applications.[66] The output of the trained agent will not be automated trading, but rather a set of actionable recommendations presented to the user through the interface, such as, "I recommend executing 20% of your order now with market orders to capture current liquidity, and placing limit orders for the next 30% at price levels X and Y to minimize impact." This approach keeps the user in control while benefiting from the AI's complex optimization capabilities.

### 5.1 Design Principles

The Natural Language Interface (NLI) is a primary gateway for users to interact with the protocol. However, in a high-leverage trading environment, the interface is not merely a tool for convenience; it is a safety-critical component where ambiguity or misunderstanding can lead to catastrophic financial loss. Its design must be grounded in established Human-Computer Interaction (HCI) research to be safe, trustworthy, and effective.

First, the design must acknowledge the significant trust deficit that exists between users and AI systems in financial contexts. Studies show that a majority of individuals still overwhelmingly trust human financial advisors over AI, especially for complex and personal decisions.[68] User trust is a multi-faceted construct influenced by socio-ethical considerations , technical features , and user characteristics .[69] Therefore, the NLI's design must prioritize building and calibrating this trust.

The design will adapt core HCI principles to this safety-critical domain.[70] Unlike consumer applications, where "ease of use" might be the primary goal, in this context, safety, clarity, and the prevention of error take precedence.[72] The interface must be designed for expert users engaging in irreversible, high-consequence actions.[72]

The following core design patterns will be implemented:

- *Explicit Confirmation*: No action that incurs financial risk (e.g., opening a position, adding leverage, submitting an order) will be executed without an explicit, multi-step confirmation process. The NLI will summarize the intended action and its most critical risk parameters (e.g., "You are about to submit a market order to buy 10 BTC-PERP with 150x leverage. Your estimated liquidation price will be $68,550.45. This action is irreversible. Please type 'CONFIRM' to proceed.").
- *Explainability*: The AI cannot operate as an opaque "black box," as this is a known barrier to user trust.[69] When the AI-assistant proposes a strategy, the NLI must provide a clear, concise rationale. For example: "I suggest splitting the order over the next 15 minutes because the order book is currently thin, and a single large market order would likely result in over 0.5% slippage."
- *Scoping and Disambiguation*: Natural language is inherently ambiguous. The NLI must be designed to robustly handle this. If a user command is imprecise, the system will not make an assumption. Instead, it will engage in a disambiguation dialogue: "How much Bitcoin would you like to buy? What level of leverage would you like to use?"
- *Progressive Disclosure:* To avoid overwhelming the user with information, the interface will present the most critical data by default while allowing the user to easily query for more granular details. For instance, the main view might show the current funding rate, but the user can ask, "Show me the historical funding rate for the past 30 days" to receive a detailed chart.
- *Trust Calibration*: A crucial safety feature is to prevent user *over-trust* in the AI. The interface must be designed to manage user expectations and communicate the inherent uncertainty of financial markets and AI predictions.[73] This can be achieved by surfacing confidence intervals for predictions or occasionally reminding the user of the limitations of the model.

By treating the NLI as a core component of the system's risk management apparatus, its design can

move beyond simple convenience to become a powerful tool for preventing user error and fostering a safe, trusted trading environment.

## 6. Performance

The protocol's performance is a direct result of its modular architecture, with each layer contributing to the overall metrics of latency, throughput, and finality.

- *Execution Latency:* The end-to-end latency for a user's trade can be modeled as the sum of several components: $L_{total} = L_{network} + L_{sequence} + L_{execute} + L_{confirm}$.
    - $L_{network}$: The network latency for the user's transaction to reach the sequencer network.
    - $L_{sequence}$: The time required for the decentralized sequencer network to reach BFT consensus on a batch of transactions.
    - $L_{execute}$: The core computational latency on the SVM L2. Leveraging Solana's parallel processing architecture and performance benchmarks [25], this component, which includes order routing, matching, and risk checks, is projected to be consistently below the 75ms target.
    - $L_{confirm}$: The network latency for the execution confirmation to return to the user. The dominant factor for the user's perceived "fast path" confirmation will be the sequencer and execution latency, which is expected to be in the low hundreds of milliseconds, competitive with centralized venues.

- *Throughput*: The system's maximum sustainable Transactions Per Second (TPS) is primarily constrained by the consensus and data propagation capacity of the decentralized sequencer network and the data availability bandwidth of the Bitcoin L1. The SVM execution layer is not anticipated to be the bottleneck, given its ability to scale with available CPU cores and process thousands of transactions per second.[75]
- *Settlement Finality:* The protocol offers two levels of finality. A fast, probabilistic finality is achieved once a transaction is included in a batch by the sequencer network and executed on the L2. This is sufficient for most trading interactions. Absolute, irreversible finality is achieved on the Bitcoin blockchain after the optimistic rollup's challenge period has elapsed without a successful fraud proof.

## 7. Conclusion

This paper has presented a comprehensive blueprint for a next-generation decentralized perpetual swaps exchange. By employing a modular, three-layer architecture, the protocol systematically addresses the execution trilemma that constrains existing platforms. The combination of a high-performance SVM execution layer, a censorship-resistant decentralized sequencing layer, and the unparalleled security of the Bitcoin blockchain for settlement creates a system that does not compromise on speed, decentralization, or security.

The key contributions of this work are fourfold. First, we have proposed a novel system architecture that synergistically integrates leading-edge technologies from different ecosystems. Second, we have formulated a rigorous mathematical adaptation of the Almgren-Chriss optimal execution model for the specific microstructure of cryptocurrency derivatives and framed it as a Markov Decision Process. Third, we have outlined a coherent risk management framework based on Conditional Value at Risk and continuous stress testing, capable of safely supporting industry-leading 150x leverage. Finally, we have established a set of HCI principles for designing a safe and trustworthy natural language interface for this complex and safety-critical financial domain.

Future research can extend this work in several promising directions. Further research into more efficient and trust-minimized multi-party fraud-proof schemes could reduce reliance on the initial set of L2 operators.[76] The AI-assisted trading agent could be enhanced with more sophisticated RL techniques, such as multi-agent reinforcement learning, to model the strategic behavior of other market participants and create more robust execution policies. Lastly, the application of formal verification methods to the protocol's smart contracts and, critically, to the logic of the NLI, could provide mathematical guarantees of system safety and correctness, further enhancing user trust and protocol security.

References

[1] Biais, B., Glosten, L., & Spatt, C. (2005). Market Microstructure: A Survey of the Literature.
*Journal of Financial and Quantitative Analysis*.
[2] Cornell University. (2025). Perpetual Futures Contracts and Cryptocurrency.
*Cornell Business*.
[3] Alexander, C., & Dakos, M. (2020). Price discovery and microstructure in ether spot and derivative markets.
*ResearchGate*.
[4] He, Z., et al. (2022). Fundamentals of Perpetual Futures.
*ResearchGate*.
[79] The Microstructure Exchange. (n.d.). A virtual seminar for market microstructure.
[5] Cornell University. (2024). The Emerging Market: Cryptocurrencies & Perpetual Contracts.
*Cornell Business Hub*.
[80] Fukasawa, M. (2018). Optimal Execution with Volume-Weighted Average Price.
*RePEc*.
[9] Gatheral, J., & Schied, A. (2011). Optimal Execution with Non-Linear Impact Functions and Trading-Enhanced Risk.
*ResearchGate*.
[15] Schnaubelt, M. (2024). Optimal trade execution in cryptocurrency markets.
*Digital Finance*.
[10] Almgren, R., & Chriss, N. (2000). Optimal Execution of Portfolio Transactions.
*University of Waterloo*.
[81] Guéant, O., & Lehalle, C. A. (2018). Optimal Execution with Volume Uncertainty.
*arXiv*.
[11] Clark, J. (2019). Optimal Execution of Large Block Orders on the NYSE and in Cryptocurrency Exchanges.
*Claremont McKenna College Theses*.
[6] CryptoAPIs. (n.d.). What Are Rollups? Understanding Blockchain's Layer 2 Scaling Solution.
[82] Gemini. (2025). Blockchain Scalability and Privacy: The Rollup Ecosystem.
*Cryptopedia*.
[83] Webisoft. (n.d.). Examples of Layer 2 Blockchains.
[30] Chainlink. (n.d.). Bitcoin Layer 2s: Scaling Bitcoin with Rollups, State Channels, and Sidechains.
*Chainlink Education Hub*.
[32] The Crypto Recruiters. (n.d.). A Guide to Layer 2 Scaling Solutions.
[84] Galaxy Digital. (2024). Bitcoin's Modular Future: The Rise of Bitcoin Layer 2s.
[36] Linus, R. (2023). BitVM: Compute Anything on Bitcoin.
*bitvm.org*.
[37] Kraken. (n.d.). What is BitVM?

*Kraken Learn*.
[85] Rootstock. (n.d.). BitVM Explained: A Guide to Smart Contracts on Bitcoin.
[38] Neptune Mutual. (2023). Unraveling BitVM: Everything You Need to Know.
*Medium*.
[86] Samara AG. (n.d.). What Is BitVM?
[87] Fidelity Digital Assets. (2024). An Overview of the Bitcoin Virtual Machine (BitVM).
[8] Astria. (2024). Astria: The Shared Sequencer Network.
[27] Espresso Systems. (n.d.). The Base Layer for Rollups.
[7] Watts, J. (n.d.). The Ultimate Guide to Sequencers in L2 Blockchains.
[28] MT Capital. (2024). MT Capital Research: Decentralized Sequencer Sector Comparative Research.
*Medium*.
[88] The Rollup. (2023). Josh from Astria on Shared Sequencers [Video].
*YouTube*.
[29] RockawayX. (2023). Decentralized Sequencer.
*Medium*.
[45] Wikipedia. (n.d.). Threshold cryptosystem.
[89] Wikipedia. (n.d.). Zero-knowledge proof.
[43] Zhang, F., et al. (2023). FairBlock: Preventing Blockchain Front-Running with Minimal Overheads.
*ResearchGate*.
[46] Bettens, L. (2023). Frontrunning Protection for Blockchains using Threshold Cryptography.
*EPFL*.
[40] Choudhuri, A., et al. (2024). A Batched-Threshold Encryption Scheme for Fair-Ordered Encrypted Mempools.
*USENIX Security Symposium*.
[44] Rejolut. (n.d.). Understanding Front Running Attack on Blockchain.
[41] McCorry, P., et al. (2022). Teaching and Experimenting with Front-Running in Decentralised Finance.
*University of Bristol*.
[42] Wang, K., et al. (2023). SoK: Mitigation of Front-Running in Decentralized Finance.
*ResearchGate*.
[90] GraphApp.AI. (n.d.). Zero-Knowledge Proofs: Implementing Privacy in Blockchain Applications.
[91] Jo, T. (2020). Zero-Knowledge Proofs in Blockchain Technology.
*University of Pennsylvania*.
[92] AEPD. (n.d.). Encryption and privacy (IV): Zero-knowledge proofs.
[93] Islam, M. R., et al. (2024). ZKBAR-V: A Zero-Knowledge Proof-Enabled Blockchain-Based Academic Record Verification System.
*PMC*.
[94] Zhang, Y., et al. (2024). A Survey on Zero-Knowledge Proofs: Theory, Implementation, and Applications.

*arXiv*.
[95] CACM. (2024). Unlocking the Potential of Zero-Knowledge Proofs in Blockchain.
[48] Boneh, D., et al. (2018). Verifiable Delay Functions.
*MDPI*.
[49] Rootstock. (n.d.). Verifiable Delay Functions.
*Medium*.
[96] Censor, Y., et al. (2024). Age-Aware Fairness in Blockchain Transaction Ordering for Reducing Tail Latency.
*ResearchGate*.
[50] TokenMinds. (n.d.). Verifiable Delay Functions (VDFs).
[97] Badrinarayanan, S., et al. (2016). FErChain: Enforcing Fairness in Blockchain Data Exchanges Through Verifiable Functional Encryption.
*ResearchGate*.
[47] Cachin, C., et al. (2023). Differential Order Fairness in Byzantine Consensus.
*arXiv*.
[98] Dynamic. (2025). SVM Chains Are Coming: Here's What to Watch.
[99] OSL. (n.d.). What is the Solana Virtual Machine (SVM)?
[100] Snap Innovations. (n.d.). Solana Virtual Machine: A Comprehensive Guide.
[101] Binance. (2024). A Look at the Top Solana Virtual Machine (SVM) Projects.
[18] QuickNode. (2024). SVM: Solana Virtual Machine 101.
[20] BingX. (n.d.). What Are the Top Solana Virtual Machine (SVM) Projects?
[33] MoonPay. (n.d.). What are optimistic rollups?
[34] CoinTracker. (n.d.). Optimistic Rollup.
[31] Trust Machines. (n.d.). What Are Rollups and How Can They Work on Bitcoin?
[102] CoinMarketCap. (n.d.). Optimistic Rollups for the Rest of Us.
[35] Chainalysis. (2024). Zero-knowledge rollups vs. optimistic rollups: An overview.
[103] a16z crypto. (2023). Making L2 Withdrawals Fast and Cheap [Video].
*YouTube*.
[76] BitVM GitHub Repository. (n.d.). BitVM.
[39] Linus, R. (n.d.). BitVM 2: Permissionless Verification on Bitcoin.
[77] Bitlayer Docs. (n.d.). BitVM Smart Contract.
[78] ZKSecurity. (2025). BitVM: Unlocking Arbitrary Computation on Bitcoin Through Circuit Abstractions.
[104] Fermat's Library. (n.d.). BitVM: Compute Anything on Bitcoin.
[36] Linus, R. (2023). BitVM: Compute Anything on Bitcoin.
*bitvm.org*.
[105] Altrady. (n.d.). The Impact of Leverage on Risk and Return in Crypto Trading.
[106] Payset. (n.d.). Risk Management in Cryptocurrency Trading.
[107] Gemini. (n.d.). A Guide to Crypto Leverage Trading.

*Cryptopedia*.

[108] NeoCryptoQuant. (2024). The Solana Architecture: A Comprehensive Deep Dive into the World's Fastest Blockchain.
*Medium*.

[109] CoinMarketCap. (n.d.). What is Solana (SOL)?
*Academy*.

[110] Streamflow. (n.d.). Solana And The POH Architecture: A Technical Deep Dive.

[23] CoinFabrik. (n.d.). Solana Blockchain Overview.

[111] Yakovenko, A. (2017). Solana: A new architecture for a high performance blockchain.
*Solana Whitepaper*.

[112] SGT Markets. (n.d.). Exploring the Solana Blockchain: A Deep Dive into SOL.

[113] Suykens, J. A. K., & Vandewalle, J. (2000). Least Squares Support Vector Machine Classifiers.
*Neurocomputing*.

[114] Ahmad, I., et al. (2018). Performance analysis of support vector machine based classifiers.
*ResearchGate*.

[115] Chorowski, J., & Zurada, J. M. (2015). Learning understandable neural networks with nonnegative weight constraints.
*Neurocomputing*.

[116] Kumar, M. A., & Gopal, M. (2018). Performance evaluation of support vector machine classification approaches in data mining.
*ResearchGate*.

[117] Duan, K. B., & Keerthi, S. S. (2005). Evaluation of simple performance measures for tuning SVM hyperparameters.
*Neurocomputing*.

[118] Ghaddar, B., & Naoum-Sawaya, J. (2018). High-Dimensional Sparse Support Vector Machines.
*SCIRP*.

[119] Columbia University. (n.d.). Financial Engineering and Risk Management.
*Coursera*.

[120] Chang, C. L., et al. (2012). Risk Management and Financial Derivatives.
*RePEc*.

[121] Chen, Y. (2024). Risk Management Strategies for Financial Derivatives: Addressing Systemic and Operational Challenges.
*ResearchGate*.

[122] Investopedia. (2024). What Is a Derivative?

[56] Rachev, S. T., et al. (2004). Risk Management and Portfolio Optimization in Volatile Markets.
*KIT*.

[123] Qiang, S., et al. (2022). A Literature Review of Portfolio Risk Management.
*Atlantis Press*.

[124] MDPI. (2021). Natural Language for Human–Computer Interaction.

[70] Palanque, P., & Paternò, F. (1998). Principles of Human Computer Interaction Design.
*Lambert Academic Publishing*.
[125] IJRASET. (2023). NLP and Human-Computer Interaction: Enhancing User Experience through Language Technology.
[126] Perera, H. (2023). Natural language for human computer interaction.
*ResearchGate*.
[71] Dovetail. (n.d.). What is Human-Computer Interaction (HCI)?
[127] Elsevier. (2021). Special Issue on Natural Language Processing for Human-Computer Interaction.
[73] UX Design. (2024). Building trust in opaque systems.
[128] Yao, Y., et al. (2025). MCP-Bench: A Comprehensive Benchmark for Evaluating Large Language Model Agents in Multi-Tool, Cross-Server Workflows.
*arXiv*.
[129] Peterfay, E. (2022). NFT Collection on Solana.
*Theseus*.
[130] Figment. (2023). Solana Inspired Networks.
[131] Belhadi, A., et al. (2024). A Survey on Distributed Ledger Technologies.
[132] DAO Labs. (2024). 2023 DAO Report.
[133] The Block Research. (2023). Digital Asset Data and Infrastructure.
[24] IJRASET. (2025). A Comprehensive Analysis of Transaction Speeds, Costs, and Real-World Applications of Solana, Ethereum, and SUI.
[134] Kumar, A., et al. (2024). Solana blockchain technology: a review.
*ResearchGate*.
[74] QuickNode. (n.d.). Latency Whitepaper: Solana.
[75] Zhang, S., et al. (2025). A Historical Analysis of Transaction Parallelism in Ethereum and Solana.
*arXiv*.
[135] IJRPR. (2024). Solana's Adaptability: An In-Depth Analysis of Its Key Features and Future Potential.
[25] Chorus One. (n.d.). Transaction Latency on Solana.
[51] Iyengar, G., & Ma, L. (2004). VaR and CVaR Optimization for Derivatives Portfolios.
*Cornell University*.
[52] MDPI. (2024). A CVaR-Based Portfolio Optimization Model for Cryptocurrencies.
[53] RSU Journals. (2022). A Comparative Study of Value at Risk (VaR) and Conditional Value at Risk (CVaR) Models.
[54] Trucíos, C., et al. (2019). Value-at-Risk and Expected Shortfall in Cryptocurrencies' Portfolio: A Vine Copula-based Approach.
*ResearchGate*.
[57] ZCU DSpace. (n.d.). Market Risk of S&P 500, Bitcoin and Ripple: VaR and CVaR Analysis.
[55] Cao, Y., et al. (2025). A Modular Simulation Framework for Cryptocurrency Portfolio Risk.
*arXiv*.

[58] GARP. (n.d.). Stress Testing: A Practical Guide.
[59] Investopedia. (2024). What Is Stress Testing in Finance and How Does It Work?
[60] International Journal of Business and Management. (2024). Stress Testing and Scenario Analysis in Risk Management.
[61] Belqadhi, A. (2016). Scenario Analysis and Stress-Testing with Entropy Pooling.
*ETH Zurich*.
[62] Federal Reserve. (2025). 2025 Stress Test Scenarios.
[136] Sarin, N., & Summers, L. H. (2024). Stress Testing: Lessons from the Banking Turmoil of 2023.
*Boston Fed*.
[137] Langley, K., et al. (2022). Theory of Mind in Human-AI Interaction.
*SCIRP*.
[72] Palanque, P., & Paternò, F. (1998). Designing User Interfaces for Safety Critical Systems.
*ACM SIGCHI Bulletin*.
[138] ACM. (2021). CHI Conference on Human Factors in Computing Systems Proceedings.
[139] Williams, R., et al. (2022). Integrating Psychological Theories into Human-Centred AI Design.
*SCIRP*.
[68] Rethinking65. (2025). For Retirement Planning, Americans Still Trust Humans Over AI.
[69] Ang, J., et al. (2023). User Trust in AI-enabled Systems: A Human-Computer Interaction Perspective.
*arXiv*.
[17] Squads. (n.d.). Solana SVM & Sealevel: The Virtual Machine That Makes Solana So Fast.
[140] Vassilyev, D. (2024). Simulation of Turbine Protocol in Solana.
*Tampere University*.
[21] Yakovenko, A. (2019). Sealevel — Parallel Processing Thousands of Smart Contracts.
*Solana*.
[141] Agadakos, I., et al. (2025). Understanding Solana's Execution Layer: RAM and Program Cache.
*arXiv*.
[142] Chainalysis. (2024). The first Solana Virtual Machine L2 on Ethereum.
[22] Hyperlane. (2024). SVM Expansion: The Landscape Beyond Solana.
*Medium*.
[63] MathWorks. (n.d.). Deep Reinforcement Learning for Optimal Trade Execution.
[64] Nevmyvaka, Y., et al. (2006). Reinforcement learning for optimized trade execution.
*Semantic Scholar*.
[67] Princeton University. (2022). Optimal Trade Execution with Reinforcement Learning.
[65] Ning, B., et al. (2021). A Deep Reinforcement Learning Framework for Optimal Trade Execution.
*ResearchGate*.
[143] Kolm, P. N., & Ritter, G. (2021). Optimal Execution Timing.
*arXiv*.
[144] Hafsi, Y., & Vittori, E. (2024). Optimal Execution with Reinforcement Learning.

*arXiv*.
[145] Shneiderman, B. (2022). Human-Centered AI.
*Oxford University Press*.
[12] Cartea, Á., et al. (2020). Enhancing the Liquidity of a Central Limit Order Book with a Market-Making Reinforcement Learning Agent.
*MDPI*.
[66] Ning, B., et al. (2020). Optimal Trade Execution Based on Deep Deterministic Policy Gradient.
*ResearchGate*.
[13] Ascione, R. (2022). Deep Reinforcement Learning for Optimal Trade Execution.
*Politecnico di Milano*.
[16] Chen, J., et al. (2018). A Partitioning Algorithm for Markov Decision Processes with Applications to Market Microstructure.
*ResearchGate*.
[14] Lim, A., & Memon, A. (2022). Deep Reinforcement Learning for VWAP Optimal Trade Execution.
*MDPI*.
[146] Gao, J., et al. (2025). Latency-Aware Market Making with Multi-Price Level Reinforcement Learning.
*arXiv*.
[19] Jito Network. (n.d.). Introduction to Solana.
[26] Jito Network. (n.d.). Why Solana for Restaking?
[147] Jump Crypto. (n.d.). A Foray into Scalability.
[148] Jump Crypto. (n.d.). Firedancer Reliability.
[149] ACM. (2022). CCS '22: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security.
[11] Clark, J. (2019). Optimal Execution of Large Block Orders on the NYSE and in Cryptocurrency Exchanges.
*Claremont McKenna College Theses*.
[36] Linus, R. (2023). BitVM: Compute Anything on Bitcoin.
*bitvm.org*.
[8] Astria. (2024). Astria: The Shared Sequencer Network.
[27] Espresso Systems. (n.d.). The Base Layer for Rollups.
[51] Iyengar, G., & Ma, L. (2004). VaR and CVaR Optimization for Derivatives Portfolios.
*Cornell University*.
[72] Palanque, P., & Paternò, F. (1998). Designing User Interfaces for Safety Critical Systems.
*ACM SIGCHI Bulletin*.
[69] Ang, J., et al. (2023). User Trust in AI-enabled Systems: A Human-Computer Interaction Perspective.
*arXiv*.