OTC Quest: An Immutable, Decentralized Protocol for Over-the-Counter Token Settlement

Otoshi

March 2026

Abstract

We propose a fully decentralized, immutable smart contract protocol for peer-to-peer over-the-counter (OTC) token settlement on EVM-compatible blockchains. The protocol enables any two parties to execute large token swaps without relying on intermediaries, centralized order books, automated market makers, or custodial escrow services. Orders are created by depositing tokens into an ownerless smart contract and settled atomically in a single transaction when a counterparty fills the order. The contract has no admin keys, no upgrade mechanism, no pause functionality, and no owner. Once deployed, the protocol operates autonomously and cannot be altered, censored, or seized by any party, including its creators. The system charges a flat 1% fee per side, supports fee-on-transfer tokens on the sell side, and enforces exact payment on the buy side to prevent economic exploitation. Identical contracts are deployed across eight EVM chains, providing a unified OTC settlement layer for the broader decentralized finance ecosystem.

1. Introduction

Over-the-counter trading accounts for a significant portion of total cryptocurrency transaction volume. Unlike exchange-based trading, OTC transactions occur directly between two parties at negotiated prices, typically for large block sizes that would cause substantial slippage on public order books or automated market makers (AMMs). Institutional investors, project treasuries, venture funds, and large token holders regularly execute multi-million dollar OTC trades to avoid market impact.

Despite the scale and frequency of these transactions, the infrastructure supporting OTC crypto trading remains remarkably primitive. The current process typically involves: (a) discovering a counterparty through private channels such as Telegram, Signal, or OTC broker networks; (b) negotiating terms manually; (c) settling via multisig wallets, centralized escrow services, or direct peer-to-peer transfers that

require one party to send first. Each step introduces trust dependencies, counterparty risk, and operational overhead.

Recent examples illustrate the scale of this problem. In March 2026, the World Foundation publicly disclosed $65 million in OTC token sales conducted with four counterparties over a single week, settled through a multisig wallet with manual coordination. This is not an exception; it is the norm. Billions of dollars in OTC crypto trades settle through trust-based mechanisms every month.

OTC Quest addresses this gap by providing a trustless, permissionless, and immutable settlement layer for OTC token swaps. The protocol eliminates every trust dependency in the OTC workflow by replacing human intermediaries with deterministic smart contract logic.

2. Problems with Current OTC Infrastructure

2.1 Counterparty Risk. In most OTC arrangements, one party must transfer assets before the other. This creates a window of vulnerability where the first sender has no guarantee of receiving the agreed-upon consideration. Escrow services mitigate this but introduce a trusted third party with custody of both sides' assets.

2.2 Custodial Risk. Centralized OTC desks and escrow providers hold user funds during the settlement process. These entities are subject to regulatory seizure, operational failure, insider theft, and smart contract vulnerabilities in their own infrastructure. Users must trust that the custodian will faithfully execute the settlement.

2.3 Slippage and Market Impact. Large orders executed on public AMMs or central limit order books suffer from price slippage proportional to order size relative to available liquidity. A $1 million swap on a pool with $10 million in liquidity can move the price significantly, resulting in unfavorable execution. OTC trades avoid this by settling at a fixed, pre-agreed price outside the public market.

2.4 Opacity and Auditability. Trust-based OTC settlements leave no verifiable public record of the terms, execution, or settlement of the trade. Disputes are resolved through social consensus or legal channels rather than deterministic onchain logic.

2.5 Censorship and Seizure. Centralized escrow services and OTC desks can be compelled to freeze funds, reverse transactions, or deny service to specific addresses. Users have no recourse if the intermediary refuses to execute a legitimate settlement.

3. Protocol Design

OTC Quest is a single, ownerless smart contract deployed identically across eight EVM-compatible chains: Ethereum, Base, BSC, Arbitrum, Polygon, Monad, Sonic, and HyperEVM. The contract implements three core functions: order creation, order filling, and order cancellation.

3.1 Order Creation. A maker initiates an OTC order by calling `createOrder()`, specifying the sell token, buy token, sell amount, buy amount, an optional designated taker address (for private deals), and an optional expiry timestamp. The maker's sell tokens are transferred into the contract at this time. For tokens with built-in transfer taxes (fee-on-transfer tokens), the contract measures the actual received amount using a `balanceBefore` / `balanceAfter` pattern and records the true deposited quantity as the order amount.

3.2 Order Filling. A taker fills an open order by calling `fillOrder()`. The contract verifies the order is open, not expired, and that the caller is authorized (either the designated taker or any address for open orders). The taker's buy tokens are transferred to the contract, the contract verifies that the full `buyAmount` was received (rejecting any underpayment from transfer taxes), deducts the 1% protocol fee from each side, and atomically distributes tokens to both parties. If any transfer fails, the entire transaction reverts. Settlement is all-or-nothing.

3.3 Order Cancellation. Only the original maker can cancel an unfilled order by calling `cancelOrder()`. The contract returns all deposited sell tokens to the maker. This function is available at any time for unexpired orders and remains available after expiry for token reclamation. No other address can access or reclaim the maker's tokens.

4. Immutability and Trustlessness

The OTC Quest contract is designed to be fully autonomous from the moment of deployment. The following properties are enforced by the contract architecture:

No Owner. The contract does not inherit from OpenZeppelin's Ownable or any access control module. There is no `owner()` function, no `transferOwnership()`, and no `renounceOwnership()`. The contract has never had an owner.

No Admin Keys. No address has elevated privileges. The deployer has no special access after deployment. The only privileged value is the fee recipient address, which is set as `immutable` in the constructor and cannot be changed.

No Pause Mechanism. The contract cannot be paused, halted, or frozen by any party. There is no circuit breaker. The protocol operates continuously and unconditionally.

No Upgrade Proxy. The contract is not deployed behind a proxy pattern. There is no mechanism to replace, modify, or extend the contract logic after deployment. The bytecode is final.

No Censorship Capability. No address can block, blacklist, or deny service to any user or token. The contract is permissionless and neutral.

5. Fee Structure

The protocol charges a flat 1% fee on each side of every trade, calculated in basis points (100 BPS = 1%). The fee is deducted at the time of order filling, not at order creation. Both the fee rate and the fee recipient address are immutable constants set at deployment.

For a trade where the maker sells 10,000 USDC for 4 ETH: the maker receives 3.96 ETH (4 ETH minus 1% fee of 0.04 ETH), and the taker receives 9,900 USDC (10,000 USDC minus 1% fee of 100 USDC). The protocol receives 0.04 ETH and 100 USDC.

6. Token Compatibility

The protocol supports any standard ERC-20 token on the sell side, including tokens with built-in transfer taxes (fee-on-transfer tokens). When a fee-on-transfer token is used as the sell token, the contract measures

the actual amount received after the tax is applied and records that as the true order amount. This prevents accounting mismatches that would cause failed settlements.

On the buy side, the protocol enforces exact payment. The contract requires that the full `buyAmount` arrives in the contract after the transfer. If a buy-side token has a transfer tax that reduces the received amount below the stated `buyAmount`, the transaction reverts. This protects makers from receiving less than their stated price. Takers using tax tokens as payment must approve sufficient allowance to cover the tax so the contract receives the full amount.

7. Security Considerations

7.1 Reentrancy Protection. All state-changing functions use OpenZeppelin's `ReentrancyGuard` modifier to prevent reentrancy attacks.

7.2 Safe Token Transfers. All ERC-20 interactions use OpenZeppelin's `SafeERC20` library, which handles non-standard return values and reverts on transfer failure.

7.3 Order Existence Validation. All functions that operate on orders verify that the order ID exists (is less than `orderCount`) before proceeding. Non-existent orders revert with `OrderNotFound`.

7.4 Atomic Settlement. The `fillOrder()` function executes all transfers in a single transaction. If any transfer fails (insufficient balance, insufficient allowance, blacklisted address, or any other reason), the entire transaction reverts. Neither party can lose funds due to a partial settlement.

7.5 Front-running Mitigation. For high-value trades, makers can specify a designated taker address to prevent front-running by MEV bots or other opportunistic actors. Only the specified address can fill the order.

7.6 Address Verification. The protocol displays full wallet and contract addresses throughout the user interface to prevent address poisoning and phishing attacks. Truncated addresses are not used.

8. Multi-Chain Deployment

The identical contract bytecode is deployed across eight EVM-compatible chains: Ethereum, Base, BSC (BNB Chain), Arbitrum, Polygon, Monad, Sonic, and HyperEVM. Each deployment uses the same verified source code, the same compiler version and settings, and the same constructor parameters. The contract is verified on each chain's block explorer, allowing anyone to inspect and audit the source code.

This multi-chain architecture allows users to select the chain that best fits their needs based on gas costs, settlement speed, and token availability, while maintaining identical security guarantees across all deployments.

9. Use Cases

9.1 Project Treasury Diversification. Protocol treasuries can sell native tokens for stablecoins or ETH at pre-negotiated prices without causing market impact.

9.2 Investor Token Sales. Early investors or foundations can execute large token sales to institutional counterparties with onchain settlement guarantees.

9.3 Private Bilateral Swaps. Two parties who have agreed on terms privately can use OTC Quest as a trustless settlement layer by creating a private order with a designated taker.

9.4 Cross-token Settlement. Any ERC-20 to ERC-20 swap at any ratio, without requiring liquidity pools, price oracles, or token listings. Particularly useful for long-tail tokens with insufficient DEX liquidity.

10. Comparison with Existing Solutions

Feature	OTC Quest	OTC Brokers	AMM/DEX	Centralized Exchange
Custody	None (contract)	Broker/Escrow	Pool	Exchange
Slippage	Zero	Zero	Variable	Variable
Admin Keys	None	Yes	Varies	Yes
Censorship Resistant	Yes	No	Varies	No
Fee-on-Transfer Support	Sell side	Manual	Limited	No
Atomic Settlement	Yes	No	Yes	No
Permissionless	Yes	No	Yes	No
Private Orders	Yes	Yes	No	No
Immutable	Yes	No	Varies	No

11. Future Chain Expansion

The OTC Quest protocol is designed to be deployed on any EVM-compatible blockchain. The current deployment spans eight chains: Ethereum, Base, BSC, Arbitrum, Polygon, Monad, Sonic, and HyperEVM. Each deployment uses identical source code, compiler settings, and constructor parameters, ensuring uniform security guarantees across all networks.

Additional EVM chains may be added in the future based on ecosystem demand, user adoption, and chain maturity. Because each deployment is an independent, immutable contract instance, expanding to new chains does not affect existing deployments. No migration, no upgrade, and no coordination is required between chains. Each chain's contract operates autonomously from the moment of deployment.

12. Conclusion

OTC Quest provides a minimal, immutable, and trustless settlement primitive for over-the-counter token trading. By removing all administrative control, upgrade mechanisms, and custodial dependencies, the protocol ensures that settlement logic is deterministic, censorship-resistant, and permanent. The contract's simplicity is its strength: three functions, no owner, no proxy, no governance. It does one thing and it does it irrevocably.

The protocol is not a replacement for decentralized exchanges, automated market makers, or sophisticated trading platforms. It is a settlement layer, a piece of infrastructure that exists at the most fundamental level of token commerce: two parties, two tokens, one atomic swap. Everything else (counterparty discovery, price negotiation, and trade communication) happens offchain. OTC Quest handles only the part that requires trust, and replaces that trust with code.
