Zond ↔ EVM Atomic-Swap Whitepaper

Two-Way Exchange of ZND ⇄ ETH

Abstract

We present a **non-custodial, two-way atomic swap protocol** enabling direct, peer-to-peer exchange of **ZND** (native Zond token) and **ETH** (or any EVM-chain token) without wrapping or relayers. Leveraging **Hashed Timelock Contracts (HTLCs)** on both chains, this design ensures either both legs of the swap execute—or neither does—eliminating counterparty risk. Though it does not introduce post-quantum cryptography on EVM, it delivers maximum trust minimization for cross-chain asset swaps today.

Table of Contents

1 Motivation

- Previous **bridge design** for ZND ⇄ EVM mints wrapped tokens that inherit the weakest chain's security.
- **HTLC-based atomic swaps** provide a *bridge-less* alternative: no mint/burn, no custody, no relayer.
- Users demand a **trust-minimized path** to acquire or exit ZND while avoiding centralized exchanges.

2 Design Goals

Goal	Rationale
Two-Way Symmetry	Either asset can initiate the swap.
Minimal Trust	No third-party signing, no liquidity escrow beyond HTLCs.
Deterministic Finality	Exactly *one* of two outcomes: successful exchange or full refund.
PQ-Aware	Keep ZND's PQ security intact; limit ETH exposure window.

3 Smart-Contract Specification

```
function lock(bytes32 H, address recipient, uint256 expiry) payable;

function claim(bytes memory S);          // requires SHA256(S) == H

function refund();                       // callable after expiry

event Locked(bytes32 indexed H, address indexed locker, uint256 amount);

event Claimed(bytes32 indexed H, address indexed claimer);

event Refunded(bytes32 indexed H, address indexed refunder);
```

4 Swap Flow

Scenario A – *Alice Swaps ETH → ZND*

Step	Actor	Chain	Action
1	Alice	ETH	Generate secret **S**, compute **H**. Call `lock{value:ETH}` with H, Bob's ETH address, T_1.
2	Bob	ETH	Poll or subscribe to `Locked(H, Alice)` event.
3	Bob	Zond	Mirrors swap: calls `lockZond(H, AliceZondAddr, T₂)`.
4	Alice	Zond	Verifies mirror; calls `claim(S)` to receive ZND, revealing **S** on-chain.
5	Bob	ETH	Reads **S** from Zond event; calls `claim(S)` to receive ETH.
6	Anyone		After expiry, unpaid HTLCs can invoke `refund()`.

Scenario B (*Bob swaps ZND → ETH*) is symmetrical with roles inverted.

5 Security Analysis

Threat	Mitigation
Secret interception	Secret revealed only after both parties locked funds; interception merely expedites Bob's claim.
Replay / double-spend	H identifies unique swap; contract state machine forbids reuse.
Timelock griefing	$T_2 < T_1$ ensures counter-party has $\Delta \geq$ block-time margin.
Quantum key theft (ETH side)	Exposure window $\leq T_1$; choose T_1 such that QC cracking within this timeframe is infeasible today.
Contract bugs	Multiple audits

6 Parameter Selection

Symbol	Description	Recommended Main-Net Value
$\Delta = T_1 - T_2$	Safety margin	≥ 360 blocks (~1 h on Ethereum)
T_2	Zond expiry	3 days (262 k Zond blocks)
T_1	Ethereum expiry	$T_2 + \Delta \approx 3$ days + 1 h

7 Edge-Case Handling

1. **Unclaimed ETH (Bob offline)** – After T_1, Alice refunds automatically; ZND she already claimed is irreversible, creating *Bob-loss* scenario. Education + UI warnings essential.

2. **Chain Re-orgs** – HTLC enforces minimum confirmations (e.g., 12 ETH blocks, 20 Zond blocks) before mirror step may proceed.

3. **Dust Swaps** – Contracts enforce minimum trade size (`minSwap = 0.01 ETH / 50 ZND`) to keep fee ratio sane.

4. **Gas Spikes** – Front-end suggests T_2/T_1 values based on live gas oracle; Δ increases during congestion.

8 Developer Stack

Layer	Tools
Zond	`go-zond`, `@theqrl/web3`, Hyp Solidity compiler 0.0.3
Ethereum	Foundry `forge`, Solidity 0.8.13, OpenZeppelin 5.x
Front-End	React + TypeScript, wagmi, `@theqrl/web3`
Testing	Hardhat, Anvil fork-tests, Zond local-devnet

9 Roadmap

Phase	Milestone
M0	Internal test-nets, unit coverage ≥ 95 %
M1	Public beta on Zond-Testnet & Sepolia
M2	Independent audit × 2
M3	Main-Net launch

10 References

1. Atomic Cross-Chain Swaps: https://arxiv.org/pdf/1801.09515
2. QRL Zond Roadmap: https://www.theqrl.org/roadmap/#project-zond/
3. QRL Weekly Update: https://www.theqrl.org/weekly/
4. go-zond GitHub: https://github.com/theQRL/go-zond/
5. Base Chain: https://base.org/
6. OpenZeppelin Contracts: https://docs.openzeppelin.com

Conclusion

The Zond ↔ Ethereum Atomic-Swap Protocol provides a **bridge-less, trust-minimized, and production-ready** path for users to exchange value across cryptographically mismatched ecosystems. By keeping ZND on its native post-quantum ledger while briefly escrow-locking ETH, the design *contains*—though cannot eliminate—quantum risk, and decisively removes the largest attack surface of wrapped-token bridges.

This document serves as the implementation blueprint for engineering teams, auditors, and community contributors aiming to deliver secure cross-chain liquidity ahead of the coming quantum era.