Mute Swap: A Zero-Knowledge Liquidity Routing Protocol

Abstract

Mute Swap is a protocol for moving value between different blockchains in a private and efficient way.
Instead of working like a traditional bridge or mixer, Mute Swap:

- Uses **AI agents** to understand what users want to do
- Uses **zero-knowledge proofs** to verify that actions are valid without revealing sensitive details
- Uses a **network of independent brokers** to execute swaps across many chains

The goal is to let users swap assets across **multiple blockchains** through a simple, intent-based interface, while reducing how much on-chain data and metadata is exposed. Mute Swap is designed as an off-chain liquidity router that can support many heterogeneous chains through partners and integrations.

---

1. Introduction

Today's DeFi experience is often:

- Complicated (multiple bridges, many steps, several wallets)
- Fragmented (different liquidity pools on different chains)
- Highly visible on-chain (making it easy to track user activity)

Users who want to move assets between chains typically have to:

1. Swap on one exchange
2. Bridge to another chain
3. Swap again into a new token

Each step costs time, fees, and increases exposure to smart contract and bridge risk.
At the same time, almost everything happens in public on the blockchain, which can:

- Reveal wallet balances

- Expose trading patterns
- Allow third parties to link addresses or behavior over time

Existing privacy tools like mixers or some shielded pools often:

- Require users to deposit into a shared pool
- Introduce some level of custodial or smart contract risk
- Can be difficult to use or understand

Mute Swap takes a different approach.
Instead of being a bridge, a mixer, or a standard DEX, it is designed as a **liquidity routing layer** that:

- Lets users describe what they want in plain language (or programmatically via AI agents)
- Turns that intent into a **zero-knowledge proof**
- Sends that proof to a **network of brokers** who execute the swap off-chain
- Delivers assets to the user through **privacy-enhancing wallet flows**

The protocol aims to:

- Reduce the number of steps for the user
- Limit public exposure of addresses and transaction links
- Avoid keeping user assets under protocol custody for longer than necessary for settlement

---

2. Architecture Overview

2.1 Legacy Cross-Chain Flow (simplified)

A typical cross-chain flow looks like this:

1. Swap token A → ETH (or another base asset) on an exchange
2. Use a bridge to move ETH to another chain
3. Swap ETH → target token on the destination chain

This can involve:

- Multiple interfaces
- Multiple signatures
- Multiple on-chain transactions
- Exposure of each intermediate step on the blockchain

2.2 Mute Swap Flow (high-level)

Mute Swap aims to compress this into a simpler pipeline:

1. **User intent:**
   The user (or an AI agent) expresses what they want, such as
   "Swap 0.5 BTC to ETH on Arbitrum."

2. **Intent encoding (Whisper AI):**
   The request is interpreted by **Whisper**, the intent engine, and transformed into a structured request.

3. **Zero-knowledge proof:**
   A **zkSNARK proof** is generated on the client side, proving that the request is valid according to the protocol rules, without exposing sensitive details like wallet addresses.

4. **Broker execution:**
   The proof is sent to a **broker mesh** (a network of independent liquidity providers). A broker verifies the proof and executes the trade flow off-chain using its own liquidity and connections (exchanges, OTC desks, etc.).

5. **Delivery to the user:**
   The user receives the resulting assets in a **privacy-enhanced wallet destination**, designed to make it harder for third parties to link the original source and final destination on-chain.

From the user's point of view, this reduces the experience to:

> "Describe what I want → approve once → receive the result."

The aim is a **simple, low-interaction user experience** that hides unnecessary complexity.

---

3. Cryptography Basics

Mute Swap relies on zero-knowledge proofs to check that swaps follow the protocol rules without revealing all underlying data.

3.1 zkSNARK Circuit Design

At a high level:

- User intent is translated into constraints that describe what a **valid swap** looks like.
- These constraints form something called a **Rank-1 Constraint System (R1CS)**.
- A **zkSNARK** lets the user create a proof that “I am following these rules” without revealing all inputs.

Key components:

- **Poseidon Hashing**
  A hash function that is efficient inside zero-knowledge circuits. It is used to build commitments and Merkle trees inside the proof system.
- **Merkle Trees & Nullifiers**
  - **Merkle trees** allow the protocol to track states or entries (like commitments) in a compact way.
  - **Nullifiers** are special values that mark certain actions as “already used” so they cannot be replayed.
    Together they help prevent double-use of the same proof or state while preserving privacy.

3.2 Proof Lifecycle (simplified)

1. **User intent:**
   Whisper reads the user’s command such as:
   “Swap 0.5 BTC to ETH on Arbitrum.”
2. **Local proof generation:**
   On the user’s side, a zero-knowledge proof is created that encodes:
   - The asset they are sending
   - The asset and chain they want to receive
   - The constraints of the routing path (e.g. allowed liquidity routes)
3. **Encrypted relay:**
   The proof is sent to the broker mesh over an **encrypted channel**.
4. **Broker verification & execution:**
   A broker verifies the proof. If valid, they execute the swap using their own liquidity pathways and deliver the resulting assets to the target destination.

During this process, the system is designed so that brokers do **not** need to see:

- The user’s original wallet address
- Extra metadata about sessions

- Identity information

What matters to them is that the proof is valid and the settlement instructions are cryptographically sound.

---

4. Whisper: The Intent Layer

Whisper is the intent processing layer of Mute Swap.

4.1 What Whisper Does

- Takes a user request in **natural language** (or from an AI agent)
- Understands what the user wants (e.g. “move these assets from chain A to chain B”)
- Translates that into **structured data** that can be turned into a zero-knowledge proof

4.2 How Whisper Runs

- Whisper is designed to run in **trusted execution environments (TEEs)** where possible, adding an extra layer of protection around processing.
- It uses a **Generalized Agent Memory Enclave (G.A.M.E.)** to keep track of context and sessions in a controlled way.

Instead of exposing raw RPC endpoints or wallet internals, Whisper:

- Hides low-level blockchain calls behind the intent engine
- Focuses on turning “what the user wants” into “what needs to be proven”

This makes it easier for:

- Human users who want a simpler interface
- AI agents and bots that operate programmatically

Whisper is intended to be an **AI-native, privacy-aware intent engine** that sits in front of the underlying zero-knowledge and liquidity layers.

---

5. Liquidity Network

Mute Swap does not function as a traditional on-chain DEX with pools.
Instead, it uses a **network of off-chain brokers**.

5.1 Broker Mesh

- Brokers are independent liquidity providers.
- They have access to liquidity across different venues (exchanges, OTC desks, aggregators).
- They receive user proofs and settlement instructions through encrypted channels.

5.2 Execution & Guarantees

The protocol design includes:

- **Attestations & bonding:**
  Brokers can stake or provide proofs of commitment to honest behavior. Misbehavior can be handled through slashing or dispute mechanisms defined at the protocol level.
- **Liquidity sources:**
  Brokers may source liquidity from a variety of venues, such as:
    - Centralized exchanges
    - OTC desks
    - Market makers
    - Other routing partners
- (Examples might include services like Godex, Bisq, Trocadore, Wintermute, or DWF Labs, depending on future integrations and agreements.)
- **Privacy alignment:**
  The actual trade execution happens off-chain on the broker side, while only minimal data (or commitments) are visible on-chain.
  There are no wrapped assets or traditional bridging contracts at the protocol core.

The aim is to give users the **effect of a cross-chain swap** without requiring them to directly interact with every underlying liquidity venue.

---

6. Cold Storage Integration

Many users hold their assets in **cold storage** (hardware wallets, offline keys) and prefer not to connect those wallets directly to online apps.

Mute Swap is designed to support this via:

- **Offline proof generation:**
  Zero-knowledge proofs can be generated using data derived from offline keys,

without exposing those keys directly to an online environment.

- **Zero-session architecture:**
  The protocol aims to avoid long-lived sessions. Users do not need persistent connections or complex session management to complete a swap.

- **Encrypted relay channels:**
  Proofs and routing instructions are submitted over secure, encrypted channels, reducing the risk of interception.

This setup is intended to let users keep their security model (e.g. cold wallets) while still being able to move assets when needed, with limited exposure.

---

7. Security & Privacy Guarantees

Mute Swap is designed as a **zero-knowledge-based liquidity router**.
Its security model rests on cryptography and protocol rules rather than trust in a single centralized party.

Key design goals:

7.1 Unlinkability

- User intents are encoded into **zkSNARK proofs**.

- **Poseidon hashing** and **Merkle-based nullifiers** help hide raw data and prevent replay attacks.

- The goal is to make it harder for outside observers to link source and destination addresses purely from blockchain data.

7.2 Limited Custody by Design

- The protocol is structured so that asset handling for settlement is done by independent brokers within a **narrow operational window**.

- The core protocol itself is not designed to hold user deposits or long-term balances.

(As always in any system, users should understand how and where their assets move and which parties are involved.)

7.3 Censorship Resistance

- Transactions and proofs travel through a **distributed set of relayers** using encrypted channels.

- This reduces reliance on any single relay or endpoint.

7.4 Front-Running Protection

- Because user orders are sent as **encrypted proofs** rather than open orders, brokers and external observers have less information about user intent before execution.
- This reduces the likelihood of front-running based on visible orderflow.

7.5 Limitations

No privacy or security system is perfect. In particular:

- Zero-knowledge proofs and encrypted routing significantly **reduce** visible information, but cannot guarantee complete anonymity in all circumstances.
- Off-chain infrastructure, user devices, and network conditions can still introduce risk.
- Users remain responsible for how they use the protocol and for protecting their own keys and devices.

7.5 Responsible Privacy: Wallet Risk Filtering

Mute Swap aims to support **privacy for legitimate users**, not to become a tool for abuse.

To support this, the protocol design includes a **wallet risk filtering module** at the broker layer:

7.5.1 How It Works (conceptually)

- Brokers can reference **cryptographic lists** derived from:
    - Public risk indicators
    - Known fraud patterns
    - External intelligence feeds or on-chain analysis tools
- Instead of exposing raw wallet addresses or identities, the system can use **hash-based membership checks** compatible with zero-knowledge techniques.
- This allows brokers to decline servicing flows that are linked to clear risk signals, while keeping normal user activity private.

7.5.2 Purpose

- Reduce the chance that the protocol is used to route funds from wallets associated with:

    - Hacks
    - Fraud
    - Clearly identified high-risk behavior
- Increase confidence for:
    - Institutional users
    - Long-term adopters
    - Ecosystem partners

The guiding principle is:

> **Mute Swap supports privacy, not impunity.**
> The aim is to give lawful users strong privacy tools while allowing the ecosystem to discourage obvious abusive patterns.

---

8. Comparative Overview

The following table gives a **high-level comparison** of different approaches.
This is illustrative rather than exhaustive.

Category	Custody Risk	Privacy Level	Cross-Chain Support	AI-Native	Examples
Bridges	Medium to High (smart contracts / bridge operators)	Typically low	Yes	No	Hop, Stargate
Mixers	Medium (shared pools)	Medium to High	No	No	Traditional mixer protocols
DEXs	Medium (on-chain pools)	Limited	Usually single-chain	No	Uniswap, Curve
Mute Swap	Limited (short-window broker handling)	Higher (zk + routing)	Yes (via brokers / partners)	Yes	This protocol

Mute Swap is intended to define a new category:
AI-native, zero-knowledge-based liquidity routing.

---

9. Roadmap

This roadmap is indicative and subject to change based on research, audits, and ecosystem feedback.

- **Q1**
    - First version of zkSNARK circuits
    - Whisper AI intent parser prototype
    - Support for a broad initial set of chains (targeting 70+ through partners)
- **Q2**
    - Broker bonding and dispute / slashing mechanisms
    - Release of a **stealth wallet SDK** for integrators and builders
- **Q3**
    - Integrations with institutional custody and compliance infrastructure where appropriate
    - Agent-to-agent routing flows (AI agents trading with each other via Mute Swap)
- **Q4**
    - Exploration of upgrades using advanced techniques (e.g. zkFHE or similar approaches) for more complex encrypted computations in the intent layer

---

10. Conclusion

Mute Swap proposes a re-design of how liquidity can move across blockchains:

- Users (and AI agents) express **what** they want, in simple terms.
- Whisper turns that into **cryptographic intents**.
- Zero-knowledge proofs ensure actions follow the rules without exposing unnecessary details.
- A distributed broker network provides **off-chain liquidity execution**.

In an environment where blockchains are transparent by default and cross-chain flows are often fragmented or risky, Mute Swap's goal is to offer:

- Simpler user experience
- Stronger privacy properties

- A modular, AI-friendly routing layer

Privacy here is treated as the **default design goal**, balanced with mechanisms that discourage abuse and support sustainable, long-term adoption.

Appendix (Outline)

This whitepaper gives a high-level overview. The following technical documents can be provided separately:

- **zkSNARK Circuit Specifications**
  Formal definitions of constraints, proof systems, and parameter choices.

- **Broker Bonding & Economic Model**
  How brokers are onboarded, how bonding works, and how disputes and slashing are handled.

- **Whisper Enclave & G.A.M.E. Framework**
  Technical details of the intent engine, trusted execution environments, and agent memory.

- **Glossary**
  Plain-language explanations of key terms such as:

  - zkSNARK
  - Nullifier
  - Poseidon hash
  - Merkle root
  - G.A.M.E. framework