CashmereLabs: Zero-slippage native omnichain middleware

Reliq
CashmereLabs S.A.
reliq@cashmere.exchange

Fust
CashmereLabs S.A.
fust@cashmere.exchange

Rhino
Independent Researcher

Roman Boehringer
CashmereLabs S.A.
roman@cashmere.exchange

Tanx
CashmereLabs S.A.
tanx@cashmere.exchange

James
CashmereLabs S.A.
james@cashmere.exchange

Abstract

In this paper, we present a novel omnichain middleware which enables zero-slippage native asset transfers. While demand for transferring assets across blockchains continues to rise, there is still no effective solution for bridging native assets directly between chains. Current cross-chain bridges face three structural limitations: (i) reliance on synthetic or wrapped tokens; (ii) dependence on market-makers or off-chain fillers; and (iii) exposure to high slippage during execution. Using fillers or liquidity bridges requires an extra filler and gas fees, which adds complexity and cannot be automated alongside the original bridge transaction. Market-maker-based liquidity bridges introduce an additional intermediary and gas overhead, adding operational complexity that cannot be automated alongside the user's original transaction. This fragmented experience often forces users to struggle with synthetic assets and incur unnecessary losses. To address this, we introduce the Cashmere algorithm—a novel middleware which leverages message-based native stablecoins. A new class of middleware that supports one-click zero-slippage direct transfers of native assets with instant and guaranteed finality through Circle's CCTP and Tether's USDT0 between any EVM and non-EVM chains.

1 Introduction

As blockchain networks continue to multiply, the cross-chain ecosystem has become increasingly fragmented. This disunity undermines liquidity efficiency, hampers interoperability, and complicates systemic risk management—concerns that have drawn growing scrutiny from both industry participants and global regulators [1].

In today's multi-chain environment, cross-chain transfers typically follow one of three models: (1) lock-and-mint bridges that issue wrapped, non-native tokens,[2, 3] or (2) market-maker or intent-based solutions that maintain separate, prefunded reserves on each chain [4, 5]. (3) message-based liquidity bridges which need to pay extra fees to LPs [6]. These methods introduce notable inefficiencies and risks. Lock-and-mint bridges and message-based liquidity bridges, in particular, have proven susceptible to security breaches—with losses exceeding $2.5 billion since 2021 [7].

CashmereLabs takes a markedly different path by combining flexible message-passing with native transfers. Leveraging native infrastructures like CCTP and USDT0, which burns native stablecoins on source chain and mints native stablecoins on destination chain directly from native stablecoin issuers. CashmereLabs sidesteps the typical lock-and-mint risks and provides zero-slippage for transfers.

However CCTP, USDT0 and other native stablecoin issuers cannot perform claim and third routing operations with a single click. CashmereLabs further refines this middleware by introducing Cashmere relayers, smart-contract and backend infrastructure which enables one-click zero-slippage native omnichain transfers, omnichain native yield aggregation, omnichain payments, omnichain AI agents, omnichain asset buys/sells.

By uniting standard-finality and sub-finality messaging, CashmereLabs directly addresses the liquidity and interoperability challenges. Its architecture mitigates security vulnerabilities, one-click operations for CCTP and USDT0, seamless native omnichain on-chain payments.

2 Background

2.1 Supported Chains, Gas Fees and Finality Times

Cashmere utilizes fast finality for CCTP V2 and USDT0 transactions, whereas CCTP V1 requires full finality before execution.

Table 1: Cashmere's domains

Source Chain	**Gas Cost**	**Average Time**
Sui	$\sim$ \$0.018	few seconds
Aptos	$\sim$ \$0.002	a few seconds
Cosmos	$\sim$ \$0.002	a few seconds
Monad	N/A	a few seconds
Avalanche	$\sim$ \$0.004	a few seconds
Ethereum L1	150K gas $\sim$ \$0.35	20 seconds
Ethereum L2s	150K gas $\sim$ \$0.003	8 seconds
Solana	$\sim$ \$0.008	a few seconds

Compared to legacy bridges, Cashmere offers significantly lower gas fees and faster transaction settlement across its supported networks.

2.2 Architecture

Cashmere's architecture comprises two tightly integrated components: (1) an on-chain smart contract system deployed across a range of EVM and non-EVM blockchains, and (2) an off-chain relayer infrastructure implemented in Golang. Together, these components enable zero-slippage, message-based native asset transfers using standards such as Circle's CCTP and Tether's USDT0 [8].

On-chain System. The smart contract stack is composed of three primary modules: the `TokenMessenger`, `MessageTransmitter` and `UsdtOFT`. These modules are responsible for securely burning assets on the source chain, emitting a cryptographically verifiable message hash, and subsequently minting native assets on the destination chain. These contracts are implemented in Solidity (EVM), Move (Aptos, Sui), and TypeScript interfaces (Solana).

In particular, EVM chains leverage a unified `CashmereCCTP.sol` contract, which abstracts over token burning, relayer fee collection, attestation tracking, and destination call execution. Non-EVM chains like Solana, Cosmos, Aptos, Sui use client libraries and custom scripts (e.g., `CashmereSend.ts`, `CashmereTransfer.move`) to interface with their native message-passing layers.

Off-chain Relayer. The relayer system monitors chain-specific logs emitted by `Cashmere` contracts. Upon detecting a valid burn event, the relayer fetches an attestation from the issuer's off-chain API (e.g., Circle's attestation service), verifies it, and triggers a minting transaction on the destination chain.

The relayer is implemented as a multi-threaded, Dockerized Go application that handles:

- Chain-specific listeners
- Attestation retrieval and verification
- Transaction submission with nonce management
- Configurable fee parameters and gas-aware execution

This design ensures atomic, fault-tolerant bridging across all supported chains, including Ethereum L1/L2, Solana, Sui, Aptos, Avalanche, and Cosmos chains.

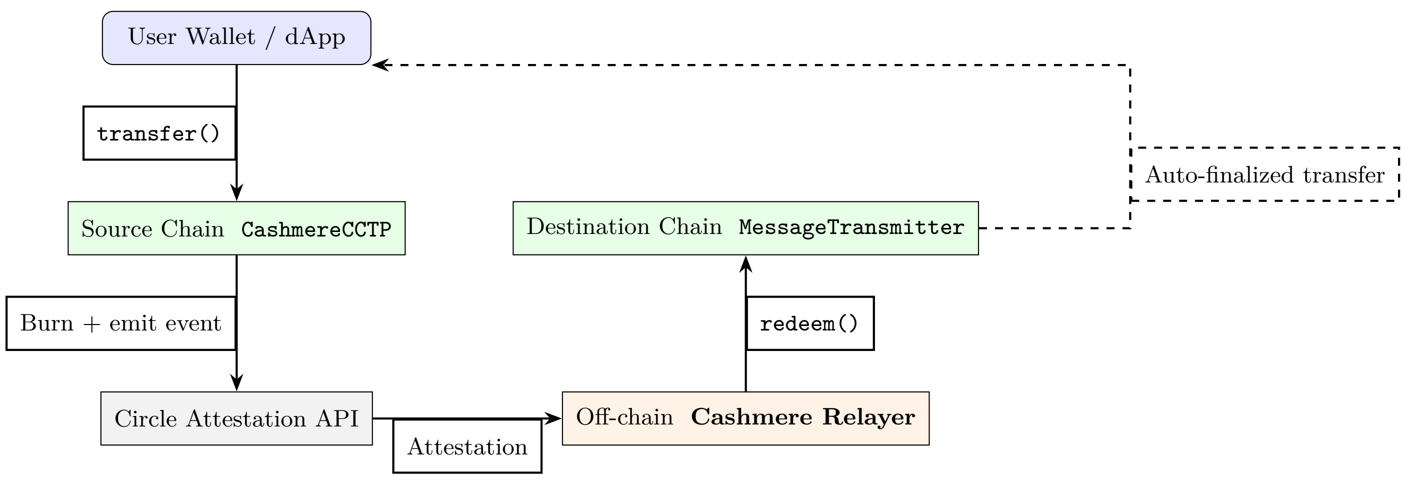

Figure 1: CashmereCCTP execution flow: from user initiation to zero-slippage native asset redemption

Cashmere Relayer can deposit native assets to yield pools, pay for on-chain payments, or buy other assets at the redeem step.

2.3 Relay Fees

Cashmere introduces a signature-based off-chain relayer gas quoting system that enables real-time, accurate, and efficient relayer gas fee collection on the source chain. This mechanism is designed to avoid the delays, costs, and complexities of on-chain or oracle-based gas estimators.

Real-time Fee Quotation via Relayer API. Cashmere's off-chain relayers expose a public API endpoint that users and frontend applications can query to obtain real-time gas fee estimates. The relayer estimates the destination chain's gas price and execution cost, computes the total required fee (including buffer), and returns it to the user along with a digital signature.

```
GET /quote?dstChainId=42161&amount=1_000_000

Response:
{
  "feeAmount": "3450000000000000",
  "expiresAt": 1715920000,
  "signature": "0xabc123..."
}
```

Listing 1: Sample API Request and Response Format

User Submission with Signed Fee, The user submits the ‘feeAmount‘, ‘expiresAt‘, and ‘signature‘ directly to the `CashmereCCTP` smart contract on the source chain when initiating the transfer. The contract uses `ECDSA.recover()` (on EVM chains) or equivalent signature verification primitives (on Sui, Solana, Aptos, etc.) to ensure that the quote was signed by a trusted relayer.

If the signature or data is invalid, the transaction is reverted. This prevents users from tampering with fee values or using expired quotes.

Efficient, Trust-Minimized Design,
This system ensures that:

- Relayers are paid exactly the amount they quoted, upfront
- Users do not need to overpay or wait for refunds
- There is no need to post Merkle roots or on-chain gas feeds
- Gas quotes are always up-to-date, minimizing failure risk

Compared to existing bridge protocols that rely on static fee tables or Merkle tree attestations, Cashmere's approach provides near-instantaneous adjustment to destination gas prices — critical in high-volatility environments like Ethereum L1 or Solana congestion.

Figure 2 illustrates the flow from user quote request to on-chain fee deduction.

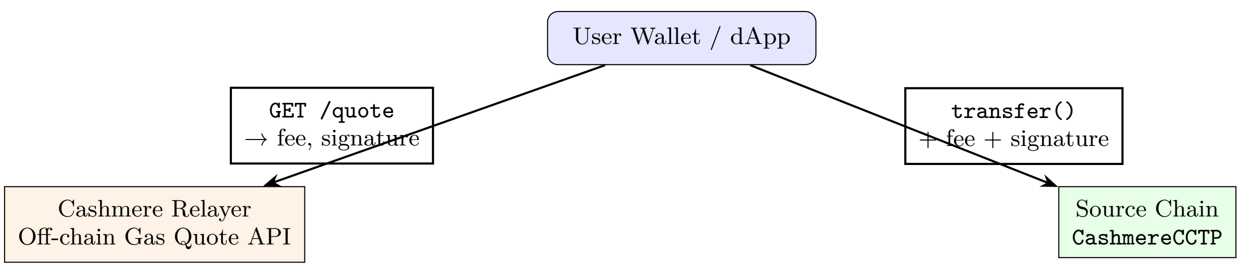

Figure 2: Real-time fee quote submission using Cashmere's off-chain relayer API

In other cross-chain protocols, relayer fees are often hardcoded or derived from on-chain gas snapshots that are only updated periodically. During periods of network volatility or gas spikes, this causes users to significantly overpay for relaying services, sometimes for extended durations. The lag in fee updates introduces inefficiency across the system and reduces cost predictability for users engaging in time-sensitive transactions. Cashmere's real-time quoting mechanism eliminates this inefficiency by providing continuously updated, signed gas fee estimates—minimizing latency losses and avoiding unnecessary overpayment.

2.4 Minimum Received Amounts

To ensure value preservation across cross-chain transfers, Cashmere enforces a **minimum received amount** mechanism that protects users from MEV, excessive relayer fees, and slippage [9, 10]. This provides a strong guarantee: users will always receive the expected amount on the destination chain—minus the relayer fee—regardless of market conditions or intermediary volatility.

Formula. At the time of initiating a transfer, the user submits a minimum amount that must be received after relayer execution. The system computes this as:

$$\texttt{minAmountOut} = \texttt{depositAmount} - \texttt{relayerFee} - \texttt{Xbps protocolFee}$$

Where:

- `depositAmount`: Total amount deposited by the user on the source chain (e.g., USDC)
- `relayerFee`: Fee signed off-chain and deducted on-chain for relay execution
- `protocolFee`: Treasury for expenses.

This mechanism ensures zero-slippage finality except gas and protocol fees.

Efficiency Guarantees.

- The gas usage on destination chains is minimized via tight calldata design and gas-efficient redemption flows
- The relayer fee is dynamically calculated using real-time gas pricing, avoiding overestimation
- No intermediary swaps or market-makers are needed, eliminating third-party friction

As a result, Cashmere consistently achieves one of the lowest total bridging costs in the cross-chain ecosystem—while maintaining guaranteed delivery semantics.

This design provides a robust settlement model that is fully composable with downstream smart contracts, cross-chain vaults, and omnichain payment flows.

3 Empirical Analysis

To validate the performance and efficiency of Cashmere, we conducted a series of real-world cross-chain transfer experiments across multiple protocols, using identical assets, routes, and timestamps. The results demonstrate that Cashmere consistently delivers higher output amounts and lower destination gas costs compared to leading alternatives such as Relay and Stargate [6].

Case Study: Solana → Ethereum USDC Transfer. In a real-world transfer of **124,089 USDC** from Solana to Ethereum:

- **Relay** delivered **123,855 USDC**, resulting in a slippage of **-234 USDC** [1]

- **Cashmere** delivered **124,066 USDC**, with a net difference of only **-23 USDC**

This represents a **90% reduction in output loss** relative to Relay, achieved through Cashmere's market-maker-free native bridge design.

Gas Cost Comparison: Ethereum → Arbitrary Chain. Using Stargate liquidity bridge, average gas cost for a cross-chain 'transfer()' transaction on Ethereum is observed at **270K–300K gas**.[2]

With Cashmere, the equivalent operation executes in **150K gas**, reflecting a **45% gas reduction** on the destination chain.

[1] `https://relay.link/transaction/0xf712a55834ed1fa45b555483616a5551679ccf9316744dc731f309364cd30f3b`

[2] `https://layerzeroscan.com/tx/0xe8c3dff4c65ad6d3367740ac0cbe8a1309a30c402db87d749d0e0d6d02e6f2d7`

Moreover, Stargate's liquidity bridge currently holds roughly $500M in committed capital, and the incentives paid out for liquidity mining impose an estimated cost of $30-50M on the protocol and token. By contrast, Cashmere facilitates zero-slippage omnichain native transfers without relying on dedicated liquidity pools; the funds that other bridges expend on incentivizing liquidity become net revenue for Cashmere. The fact that Stargate and its peers are now pivoting toward alternative approaches—such as Hydra and USDT0—serves to validate the efficiency and scalability of Cashmere's message-based, non-liquidity omnichain bridging model.

Macro Impact: For an on-chain user bridging $1M in assets monthly, we estimate:

- **$3,000+ per month in hidden costs** on other protocols due to slippage and inflated relayer/gas fees
- **0.00% market-maker spread** on Cashmere due to native asset delivery without market-maker execution
- **MEV protection**: Cashmere ensures guaranteed outputs on the destination chain, preventing sandwiching or frontrunning of cross-chain execution

3.1 Results

For the other interoperability providers baseline and the Cashmere, we analyzed how the surplus value in USD (always evaluated at the spot prices at the simulated timestamp) develops over time [11].

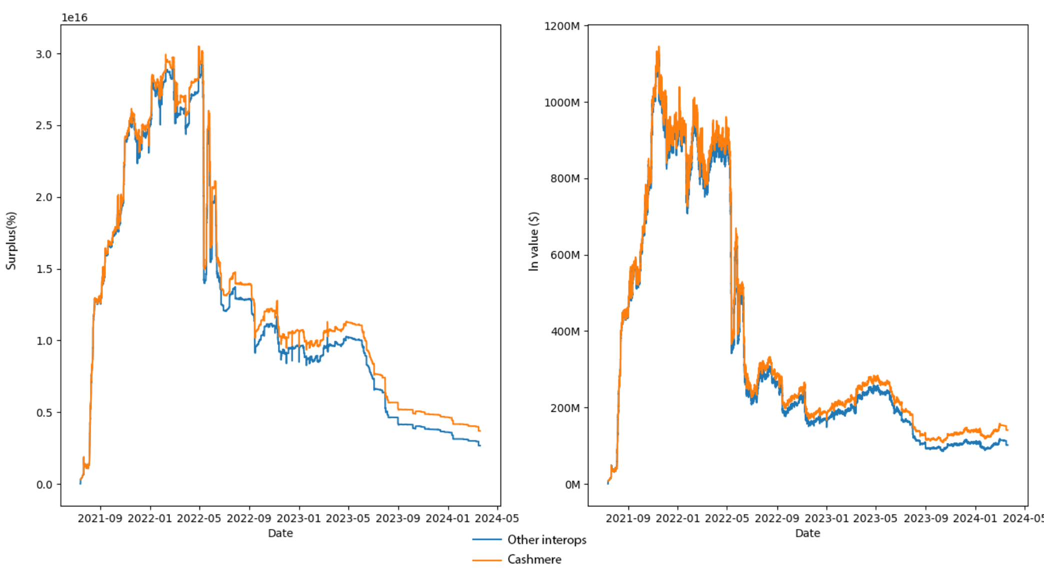

Figure 3: Value surplus over time comparing Cashmere to other bridge protocols.

These results are presented in Figure 3. Cashmere always has a surplus value that is greater than or equal to the other interoperability providers. This was to be expected because it is enforced by design. This difference results in a higher output during the whole simulation, which can be seen in the left chart.

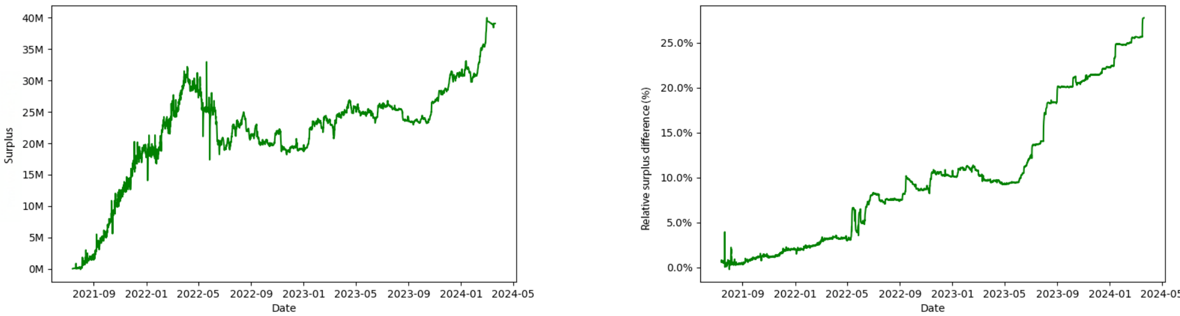

Figure 4: Absolute (left) and relative (right) surplus generated by Cashmere transfers.

In Figure 4, we analyze this difference in more detail. In the left chart, we present the absolute surplus that Cashmere generates for the cross-chain users over the simulated time span. The surplus is increasing most of the time, with the steepest changes during times of high gas volatility. This relative cross-chain transfer output surplus increases over the time span of three years to roughly 25%, i.e. an annualized surplus of roughly 7.7%.

4 Conclusion

Cashmere represents a fundamental shift in how cross-chain value transfer is executed. By leveraging native assets and real-time relayer infrastructure, it eliminates slippage, reduces fees, and achieves composability that legacy bridges and liquidity networks cannot offer. Our empirical analysis demonstrates measurable cost savings and improved reliability across EVM and non-EVM chains alike.

This architecture is not only more efficient—it is more secure, more scalable, and more aligned with user expectations. With Cashmere, users can initiate transfers, payments, and native asset interactions across blockchains with guaranteed outcomes and no market-maker dependency.

As cross-chain demand continues to grow, Cashmere is positioned to become a critical layer in the infrastructure stack of omnichain applications, simplifying integrations while delivering best-in-class performance. We believe this approach sets a new standard for usability and security in the interoperability space.

References

[1] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. Sok: Communication across distributed ledgers. In *Financial Cryptography and Data Security*, pages 3–36. Springer, 2021.

[2] Ryan Zarick, Bryan Pellegrino, Isaac Zhang, Thomas Kim, and Caleb Banister. Layerzero. *https://layerzero.network/publications/LayerZero_Whitepaper_V2.1.0.pdf*, 2023.

[3] Axelar Network. Connecting applications with blockchain ecosystems. *https://axelar.network/axelar_whitepaper.pdf*, 2021.

[4] Sebastian Doerr Frederic Boissay, Giulio Cornelli and Jon Frost. Blockchain scalability and the fragmentation of crypto. *https://www.bis.org/publ/bisbull56.pdf*, 2022.

[5] Mallesh Pai Rice University Tarun Chitra, Kshitij Kulkarni and SMG. An analysis of intent-based markets. *https://baincapitalcrypto.com/wp-content/uploads/2024/03/An-Analysis-of-Intent-Based-Markets.pdf*, 2024.

[6] Ryan Zarick and Caleb Banister Bryan Pellegrino. Stargate. *https://www.dropbox.com/s/gf3606jedromp61/Delta-Solving.The.Bridging-Trilemma.pdf?dl=0*, 2022.

[7] Chainalysis Research Team. A comprehensive survey of cross-chain bridge hacks 2021–2023. *https://www.chainalysis.com/blog/2023-bridge-hacks/*, 2023.

[8] Jonathan Lim Walker, Mike. Cctp. *https://github.com/circlefin/evm-cctp-contracts/blob/master/whitepaper/CCTPV2_White_Paper.pdf*, 2025.

[9] Andrea Canidio and Robin Fritsch. Arbitrageurs' profits, lvr, and sandwich attacks: batch trading as an amm design response. *arXiv preprint arXiv:2307.02074*, 2023.

[10] Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621, 2015.

[11] Jason Milionis, Ciamac C Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046*, 2022.