Single-Sided Liquidity Pools for FX-Linked Stablecoins on Kaia

HE ZHILI

November 30, 2025

1 CONCEPT OVERVIEW

Single-Sided Liquidity (SSL) is a DeFi mechanism designed to lower entry barriers for capital provision. Unlike traditional Automated Market Makers (AMMs) such as Uniswap V2, which require Liquidity Providers (LPs) to deposit a pair of assets of equal value (e.g., 50% USDT and 50% ETH), SSL allows LPs to deposit a single type of asset into a liquidity pool. This approach mitigates the risk of "Impermanent Loss" (IL) often caused by price divergence between paired assets in standard AMMs. To achieve this, single-sided pools separate the accounting of each asset, typically utilizing an **Asset-Liability Management (ALM)** model to track solvency.

The industry has seen successful implementations of this concept. For instance, **Platypus Finance** on Avalanche utilizes a "Coverage Ratio" mechanism where assets are pooled together, but liabilities are tracked individually, allowing for zero-slippage swaps when the pool is balanced. Similarly, **Curve V2**, while often supporting multi-asset pools, employs stableswap invariant logic that serves as a foundation for concentrating liquidity around a specific price peg or Oracle rate.

2 APPLICATION TO FX-LINKED STABLECOINS

The Foreign Exchange (FX) market represents the largest financial market globally, yet its on-chain liquidity remains highly fragmented. Current DeFi solutions typically force LPs to hold exposure to currencies they may not desire. For example, a US-based investor providing liquidity for a KRW/JPY pool would traditionally be required to hold both KRW and JPY, exposing them to unwanted FX risk. Furthermore, liquidity is currently split across hundreds of isolated trading pairs (e.g., USD-KRW, USD-JPY, KRW-JPY), diluting the market depth.

To address these inefficiencies, we propose applying Single-Sided Liquidity on the **Kaia blockchain** to create a **"Global FX Hub."** In this model, LPs deposit only their native currency—users in Korea deposit KAI-KRW, while users in the US deposit KAI-USD. The protocol then aggregates these deposits into a shared liquidity reservoir. Consequently, traders can execute *Any-to-Any* asset swaps (e.g., KAI-KRW to KAI-JPY) using this aggregate liquidity, with routing determined by Oracle pricing. This design significantly maximizes capital efficiency, as KAI-USD liquidity is not locked into a specific pair but facilitates trades against any supported asset in the pool.

3 Proposed Design Architecture

3.1 Core Mechanism: The "Coverage Ratio" Model

Instead of relying on the constant product formula ($x \cdot y = k$), this design employs an Oracle-guided pricing model with a dynamic slippage penalty based on the **Coverage Ratio (CR)**. The Coverage Ratio is defined as the total cash in the pool divided by the total liabilities owed to LPs. The system targets an equilibrium where the CR equals 100%.

The slippage logic functions to maintain this equilibrium. If a user sells KAI-USD into the pool, thereby increasing the cash reserves and pushing the USD Coverage Ratio above 100%, the protocol offers a slight premium to encourage this rebalancing action. Conversely, if a user buys KAI-USD from the pool, decreasing cash reserves and dropping the CR below 100%, the protocol applies a slippage fee to discourage actions that drain liquidity.

3.2 Pricing & Oracles

Unlike stablecoins pegged to the same fiat currency, FX rates fluctuate constantly. Therefore, the protocol cannot assume a 1:1 exchange ratio. The system relies on high-fidelity Oracles, such as Chainlink or Pyth deployed on Kaia, to fetch real-time off-chain FX rates (e.g., 1 USD = 1300 KRW). The final execution price for a trade is calculated by taking the Oracle Price and adjusting it by the dynamic slippage fee derived from the Coverage Ratio model.

3.3 Kaia Compatibility & Integration

The proposed architecture is specifically optimized for the Kaia ecosystem. It supports the **KIP-7** token standard, ensuring seamless integration with existing wallets and dApps. Crucially, Kaia's low gas fees enable frequent Oracle updates, which are essential for maintaining accurate FX pricing on-chain. Additionally, the protocol leverages Kaia's immediate transaction finality to prevent "Free Option" arbitrage strategies, where traders might otherwise exploit the latency between trade submission and block confirmation.

4 Risks, Limitations & Trade-offs

4.1 Oracle Latency & Toxic Flow

A primary risk in this architecture is Oracle latency. If the real-world FX market moves faster than the on-chain Oracle updates, arbitrageurs can trade against "stale" prices, effectively extracting value from LPs. To mitigate this, the protocol should implement a **"Volatility Circuit Breaker"** that temporarily pauses trading if the off-chain price moves significantly (e.g., $> 0.5\%$) within a short timeframe. Furthermore, the system will capitalize on Kaia's fast block times to update Oracles more frequently than is typically possible on Ethereum Layer 1.

4.2 The "Bank Run" Risk (Liquidity Crunch)

Single-sided pools face a unique liquidity risk known as a "Bank Run." If market sentiment shifts and all users attempt to withdraw KAI-USD (the stronger currency) while the pool is filled with KAI-KRW (a potentially weakening currency), the Coverage Ratio for USD could drop to dangerous levels. To address this, the protocol incorporates dynamic withdrawal limits where fees increase as the Coverage Ratio drops. Additionally, an algorithmic interest rate model will automatically increase the APY for KAI-USD deposits when liquidity is low, incentivizing new capital inflows to restore balance.

4.3 Assumption of Peg

This design operates under the assumption that the underlying stablecoins (KAI-USD, KAI-KRW) maintain their peg to their respective fiat counterparts. However, there is a trade-off: if an underlying stablecoin significantly de-pegs, the liquidity pool could inadvertently absorb the depreciating asset. To protect the protocol, an emergency shutdown trigger must be included, which halts operations if the stablecoin itself deviates beyond a safe threshold (e.g., deviating from $1.00).