Technical Documentation: Velodrome Integration

Introduction

Decentralized Finance (DeFi) platforms utilize liquidity pools, which are at the core of their operations. This technical documentation offers a deep dive into two main types of liquidity pools: Stable/Volatile Pools and Concentrated Liquidity Pools. This document will cover smart contracts, processes for adding and removing liquidity, key differences between pool types, and example workflows.

Pool Types

- **Stable/Volatile Pools**
  Stable Pools are designed for assets that maintain a stable value relative to each other (e.g., USDC/DAI).
  Volatile Pools are for assets with fluctuating prices (e.g., ETH/DAI).
- **Concentrated Liquidity Pools**
  Concentrated Liquidity Pools enable liquidity providers to allocate liquidity within specific price ranges. Non-Fungible Tokens (NFTs) represent positions and allow for custom liquidity ranges and individual fee accrual.

Required Contracts

- **Stable/Volatile Pools Contracts**
  Pool.sol: Core contract for a token pair's liquidity pool.
  Router.sol: Main contract for user interactions (adding/removing liquidity and swapping tokens).
  Gauge.sol: Contract for staking Liquidity Provider (LP) tokens to earn additional rewards.
  Factory.sol: Contract responsible for creating new pools.
- **Concentrated Liquidity Pools Contracts**
  CLPool.sol: Core pool contract with concentrated liquidity features.
  CLPoolManager.sol: Manages positions (liquidity allocations) in CLPools.
  CLGauge.sol: Contract for staking position NFTs to earn rewards.
  CLFactory.sol: Contract for creating new concentrated liquidity pools.

Adding Liquidity

- **Stable/Volatile Pools**
  Contract: Router.sol
  Function: addLiquidity
  Parameters: tokenA (address), tokenB (address), stable (bool), amountADesired (uint256), amountBDesired (uint256), amountAMin (uint256), amountBMin (uint256), to (address), deadline (uint256)
  Prerequisites: Token Approvals, Pool Existence
- **Concentrated Liquidity Pools**
  Contract: CLPool.sol
  Function: mint
  Parameters: recipient (address), tickLower (int24), tickUpper (int24), amount (uint128), data (bytes)
  Additional Function:
  Contract: CLPoolManager.sol
  Function: createPosition
  Parameters: pool (address), tickLower (int24), tickUpper (int24), amount0Desired (uint256), amount1Desired (uint256), amount0Min (uint256), amount1Min (uint256)
  Prerequisites: Token Approvals, Tick Range Calculation

Removing Liquidity

- **Stable/Volatile Pools**
  Contract: Router.sol
  Function: removeLiquidity
  Parameters: tokenA (address), tokenB (address), stable (bool), liquidity (uint256), amountAMin (uint256), amountBMin (uint256), to (address), deadline (uint256)
  Prerequisites: LP Token Approval, Unstake from Gauge
- **Concentrated Liquidity Pools**
  Contract: CLPool.sol
  Function: burn
  Parameters: tickLower (int24), tickUpper (int24), amount (uint128)
  Additional Function: collect
  Parameters: recipient (address), tickLower (int24), tickUpper (int24), amount0Requested (uint128), amount1Requested (uint128)
  Prerequisites: Position Ownership, Unstake from CLGauge

Key Differences Between Pool Types

Feature	Stable/Volatile Pools	Concentrated Liquidity Pools
Position Management	Fungible LP tokens, Single pool-wide liquidity position, Interactions through Router contract	NFTs represent individual positions, Customizable price ranges, Direct interactions with CLPool contract
Liquidity Addition	addLiquidity via Router, Automatic price calculation, Fixed price ranges	mint via CLPool, Manual selection of tick ranges, Multiple positions with different ranges
Fee Collection	Fees accumulated in pool and increase LP token value, Collected upon liquidity withdrawal	Fees tracked per position, collect function to claim fees

Example Workflows

- **Stable/Volatile Pool Workflow**
  - Adding Liquidity: Approve Tokens, Add Liquidity, Stake in Gauge (Optional)
  - Removing Liquidity: Unstake from Gauge (If Staked), Approve LP Tokens, Remove Liquidity
- **Concentrated Liquidity Pool Workflow**
  - Adding Liquidity: Approve Tokens, Calculate Tick Range, Mint Position, Stake in CLGauge (Optional)
  - Removing Liquidity: Unstake from CLGauge (If Staked), Burn Liquidity, Collect Fees**Technical Workflow**
- **Stable/Volatile Pool Workflow**
  - Adding Liquidity: Approve Tokens → Add Liquidity → Stake in Gauge (Optional)
  - Removing Liquidity: Unstake from Gauge (If Staked) → Approve LP Tokens → Remove Liquidity
- **Concentrated Liquidity Pool Workflow**
  - Adding Liquidity: Approve Tokens → Calculate Tick Range → Mint Position → Stake in CLGauge (Optional)
  - Removing Liquidity: Unstake from CLGauge (If Staked) → Burn Liquidity → Collect Fees

Contract Interactions Flow

- User: Approves tokens, Initiates liquidity addition or removal.
- Router/CLPool: Receives function calls, Interacts with pool implementations.
- Pool/CLPool Implementation: Executes core logic, Handles token transfers and liquidity management.
- Token Transfers: Tokens moved between user and pool, LP tokens or position NFTs issued.
- LP Tokens/Position NFTs: Represent user's pool share, Can be staked.
- Gauge/CLGauge Staking (Optional): Users stake LP tokens or position NFTs, Earn extra rewards.