Visor uses a smart contract-based algorithm to manage liquidity in its liquidity pools. This algorithm is designed to maintain a specific ratio of assets in each pool based on market conditions. The smart contract automatically adjusts the amount of liquidity provided by each pool to ensure that this ratio is maintained.

The specific ratio maintained by Visor's liquidity management algorithm is determined by the pool's target weight. The target weight is a number between 0 and 1 that represents the desired percentage of each asset in the pool. For example, if a pool has a target weight of 0.5 for ETH and a target weight of 0.5 for USDC, then the smart contract will aim to maintain a 50/50 ratio of ETH and USDC in the pool.

To achieve this, the smart contract periodically calculates the actual weight of each asset in the pool. It then compares this actual weight to the target weight and adjusts the amount of liquidity provided by each asset accordingly. If the actual weight of an asset is higher than the target weight, then the smart contract will reduce the amount of liquidity provided by that asset. Conversely, if the actual weight of an asset is lower than the target weight, then the smart contract will increase the amount of liquidity provided by that asset.

Let's now look at some illustrative examples of how this works in practice.

Example 1: ETH-USDC Pool

Suppose we have a liquidity pool on Visor that contains ETH and USDC. The pool has a target weight of 0.5 for each asset. Initially, the pool contains 10 ETH and 10,000 USDC, resulting in a weight of 0.5 for each asset.

Over time, the price of ETH increases relative to USDC. As a result, the actual weight of ETH in the pool increases to 0.6, while the actual weight of USDC decreases to 0.4.

The smart contract then calculates the difference between the actual weight and the target weight for each asset. In this case, the difference for ETH is 0.1 (0.6 - 0.5) and the difference for USDC is -0.1 (0.4 - 0.5).

Based on these differences, the smart contract reduces the amount of liquidity provided by ETH and increases the amount of liquidity provided by USDC. This helps to maintain the target weight of 0.5 for each asset in the pool.

Example 2: ETH-DAI Pool

Suppose we have another liquidity pool on Visor that contains ETH and DAI. The pool has a target weight of 0.7 for ETH and 0.3 for DAI. Initially, the pool contains 5 ETH and

10,000 DAI, resulting in a weight of 0.33 for ETH and 0.67 for DAI.

Over time, the price of ETH decreases relative to DAI. As a result, the actual weight of ETH in the pool decreases to 0.2, while the actual weight of DAI increases to 0.8.

The smart contract then calculates the difference between the actual weight and the target weight for each asset. In this case, the difference for ETH is -0.5 (0.2 - 0.7) and the difference for DAI is 0.5 (0.8 - 0.3).

Based on these differences, the smart contract reduces the amount of liquidity provided by DAI and increases the amount of liquidity provided

Visor Finance is one example of a protocol for automated liquidity provision management that aims to optimize LP returns. Users can provision their assets to be managed by ‘Supervisors’ that execute Uniswap v3 liquidity position management strategies based on chosen conditions and thresholds. Visor is integrated with many liquidity pools on Uniswap and utilizes a variety of market-making strategies to maximize returns while minimizing impermanent loss.

In order to automatically execute these liquidity management strategies, Visor’s smart contract needs to be notified based on external conditions and thresholds. Visor integrated Chainlink Automation to trigger certain on-chain liquidity management functions such as reinvesting fees, opening and closing limit orders, and setting price ranges. With support from Chainlink Automation, liquidity management strategies in Visor can run in an automated and reliable manner.

Adding the strong reliability guarantees of Automation to Visor’s feature set decreases the complexity for third parties running a Supervisor and empowers strategists to focus on their fundamental task—developing advanced liquidity management strategies that maintain the highest possible asset utilization for Uniswap v3 LPs.

Another example comes from Pickle Finance, which is using Chainlink Automation in its Pickle Jars product to help automatically manage capital-efficient LP positions on Uniswap v3. Chainlink Automation helps automatically rebalance users’ LP positions to help ensure they always remain in range and collect maximum LP fees. This helps enable Pickle Jars to maximize returns for users without requiring any manual intervention or dependency on centralized scripts.

AMMs pool liquidity by enabling users to deposit their tokens for a share of trading fees, and liquidity pools are automatically rebalanced by the AMM according to supply and demand. Though AMMs are highly useful tools, one limitation is that they do not feature order books, meaning users are unable to set limit orders that automatically buy or sell assets at their desired target price. Instead, traders are forced to wait for their desired price and market buying the asset once it is reached, which is extremely time-consuming and inefficient.