



Cashmere Labs

An Omnichain  
MEV Resistant  
DEX Aggregator & Stableswap

**An Omnichain DEX Aggregator & Stableswap**

# Agenda

- The problem
- The solution
- Single-side liquidity system
- Compensation ratio system
- Slippage system
- MEV attacks & Precautions
- Value Proposition
- Business Model
- Security Measures
- Roadmap
- Tokenomics

# The problem

## Problem

- **1- L2** networks in the market don't have sufficient liquidity on stablecoins also have security vulnerabilities. The **user has to pay high slippage** for a high number of swaps and trades.
- **2-** When users switch from one network to the other network, **users have to use bridge coins** other than the native coins of the networks, and this is **insecure**.
- **3-** There is no decentralized dapp in the market that uses **interoperability messaging protocol** to communicate between chains to swap any asset between any chain **without bridging**.

## Challenge

- **1-** The invariant function logic is inefficient for stableswap and can be abused. Insufficient liquidity reduces the trading volume on the network, which can be manipulated by attackers.
- **2-** When bridging assets, security issues may arise on bridged tokens and user balances may suffer.
- **3-** Most cross-chain swaps process 1 transaction in over 10 minutes. But Cashmere aims to reduce that time by integrating to Layer0 for swap functions.

## Negatives

- According to the LUNA-UST collapse, algorithmic **stablecoins should be swapped natively** with solid liquidity and securely.
- Hackers can attack and de-peg these stablecoins if a large portion of their liquidity is in a different network by bridged assets and their liquidity is bottlenecked while being transported, giving attackers an opportunity.
- However, if the liquidity of the attacked stablecoins were not on their bridged assets and **if their liquidity were native in their own network, the risk of these attacks would be much lower**.

# The solution

## 1-Low slippage

- Single-side AMM
- Single variant marginal slippage function
- Compensation Ratio
- Cross-chain aggregator

## 2-Efficient for arbitrage

- High trade volume
- Positive slippage
- Low swap fee (Haircut)

## 3-Convenient Liquidity

- Satisfactory pool Incentives with **generating real value**
- Cashmere DAO and **Reasonable Lock System**
- Native Cross-chain swaps with **interoperability messaging protocol**
- +80% of stablecoin liquidity should be on its native network.

# Cashmere Goals

## 1-Short term goals

- **Providing solid liquidity** to the **L2** Network's Stablecoin market in their native networks
- **Protect** stablecoins from **de-peg** to increase utility on their native network.
- Increase the prevalence of the stablecoins

## 2-Long term goals

- **Extending stablecoin liquidities** and prevalence to all L1 and **L2 networks**.
- **Strengthening** stables liquidity in **L1 & L2s**, facilitating its migration to other networks.
- Being the **technology solution partner of L2s**

## 3-Value creation

- **Generating real value** by distributing profits from the platform's fees to veCSM holders.
- **Increasing company value** as stablecoins grow by providing solid liquidity to stablecoins.

- Swap any asset between **any** **chain** without **bridging assets**
- Cross-chain** stableswap
- Inter-chain** stableswap
- Decentralized **emission rates**
- controlling by** DAO



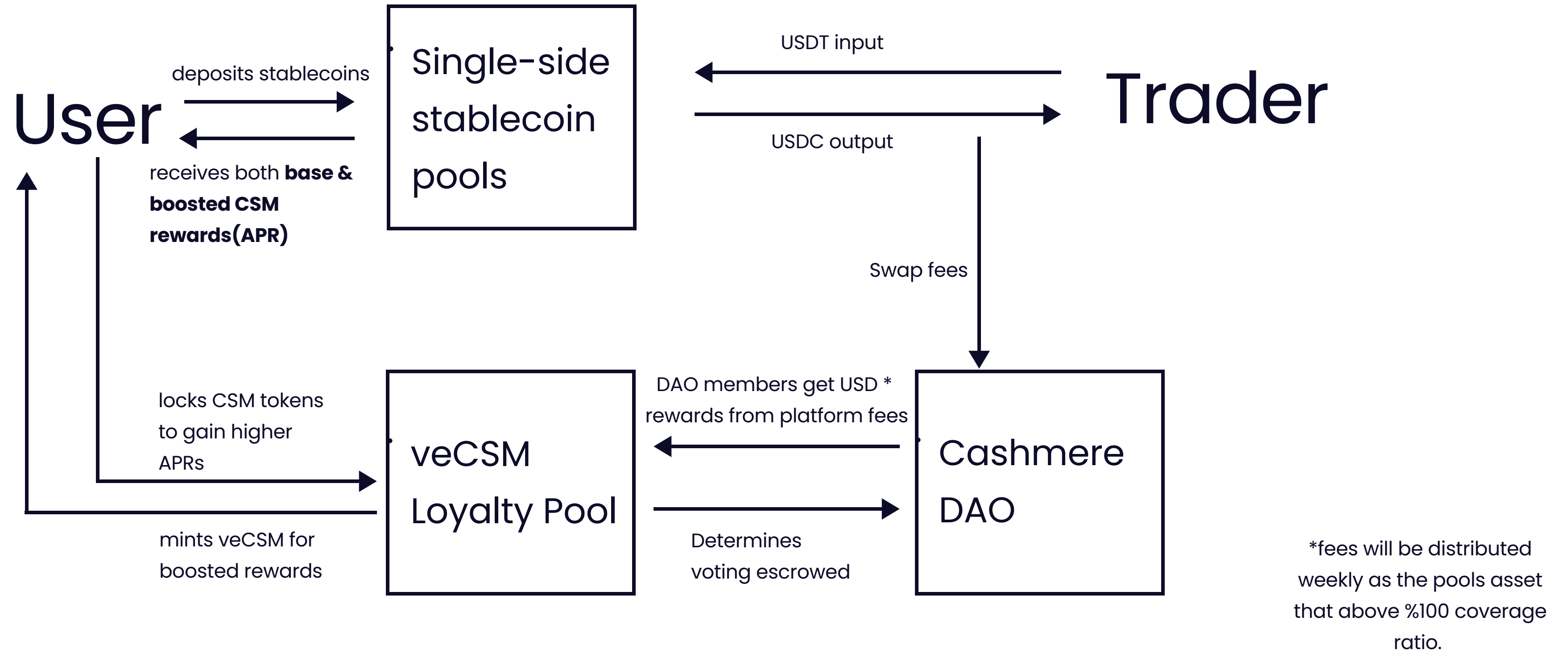
**All in  
one**

**Your unique  
cross-chain  
stableswap & asset  
aggregator platform  
goes here.**



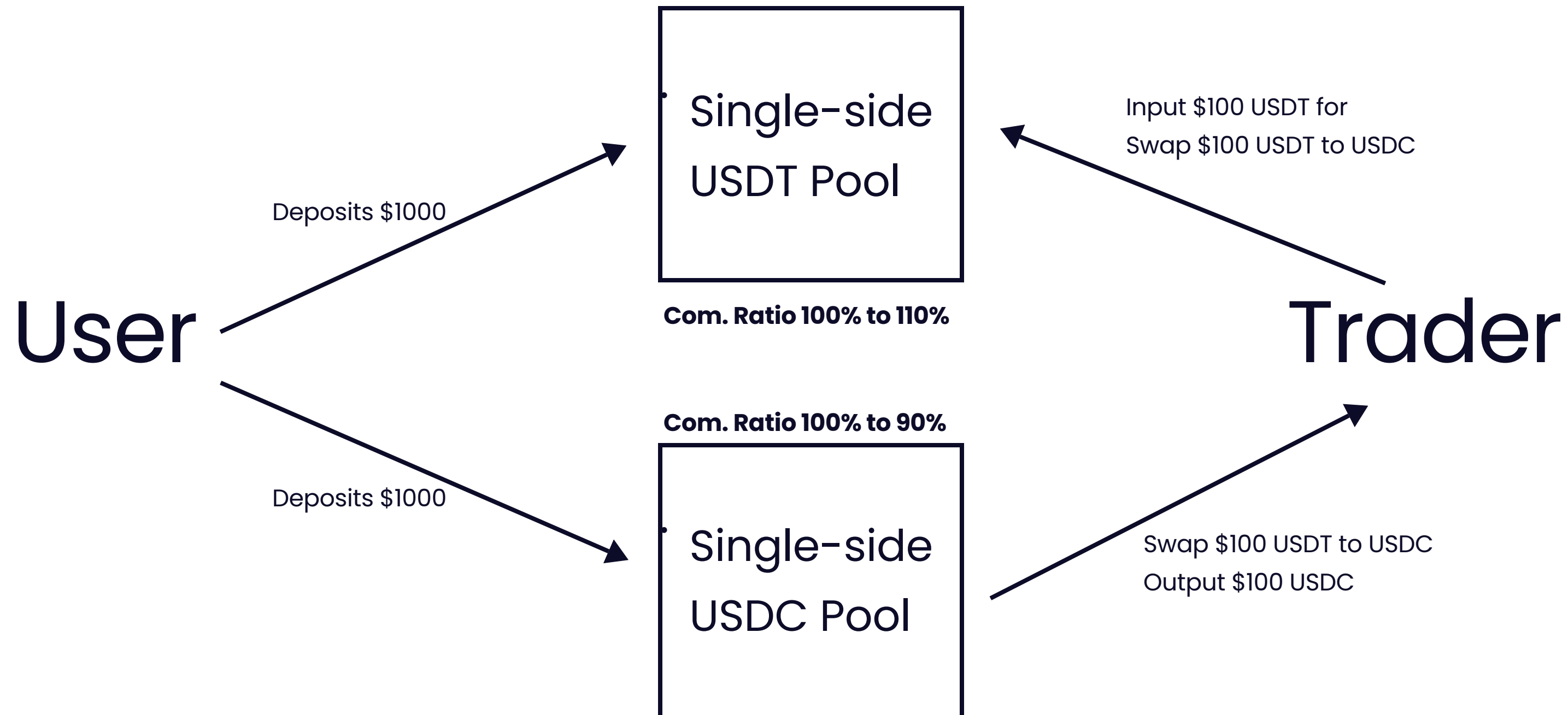
# Single-side liquidity system

Haircut: 0.04%



# Compensation Ratio System

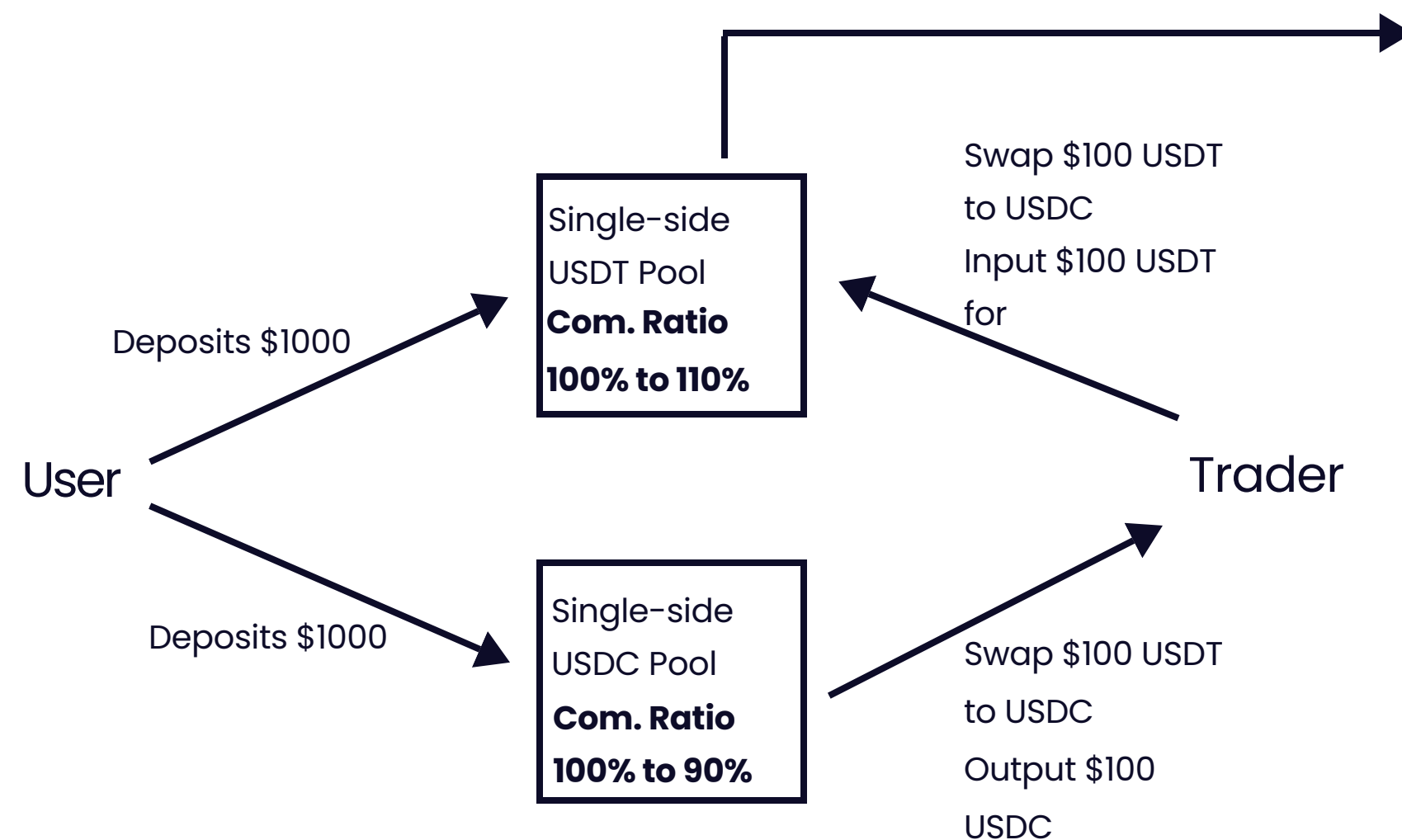
\*Initial Com. Ratio 100% for both pools



• Compensation Ratio = Asset / Liability

# Slippage & Positive Arbitrage System

\*Initial Com. Ratio 100% for both pools



We take the compensation ratio of USN at 0.9 and USDT at 1.1. Working this out, we'd get:

**USN:**

$$f'(0.9) = -\frac{0.00002 * 7}{0.9^8} = 0.0325\%$$

**USDT:**

$$f'(1.1) = -\frac{0.00002 * 7}{1.1^8} = 0.0065\%$$

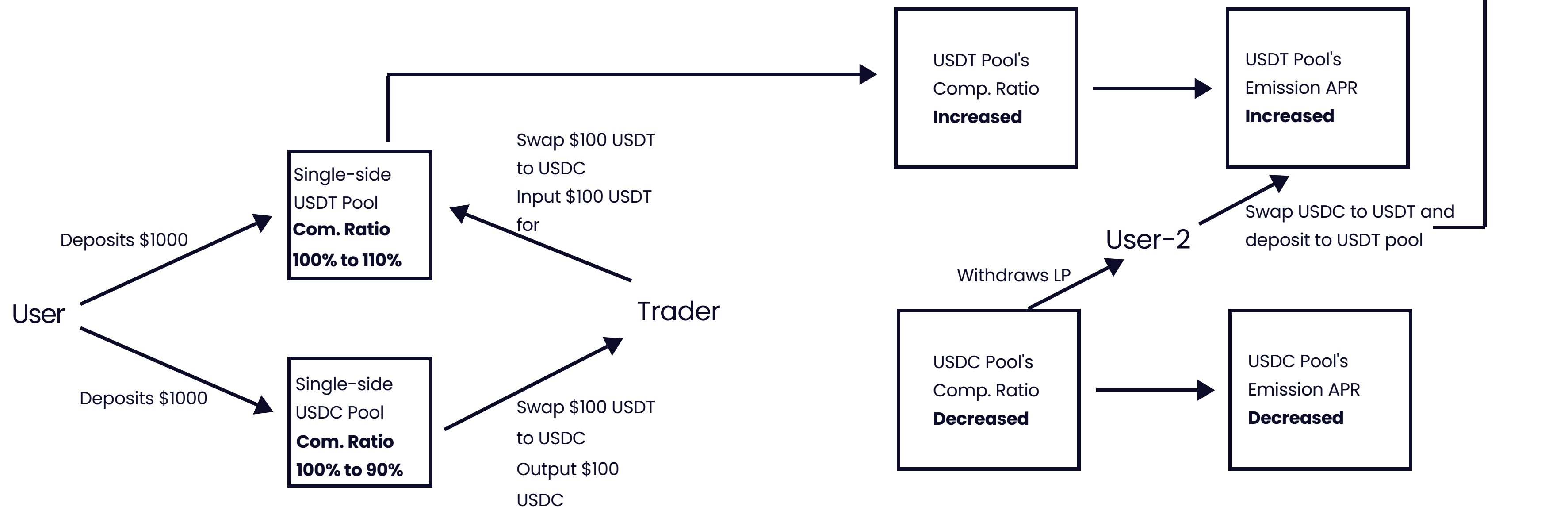
Hence we have

$$C_{USN \rightarrow USDT} = 0.0325\% - 0.0065\% = 0.026\%$$

- Compensation Ratio = Asset / Liability

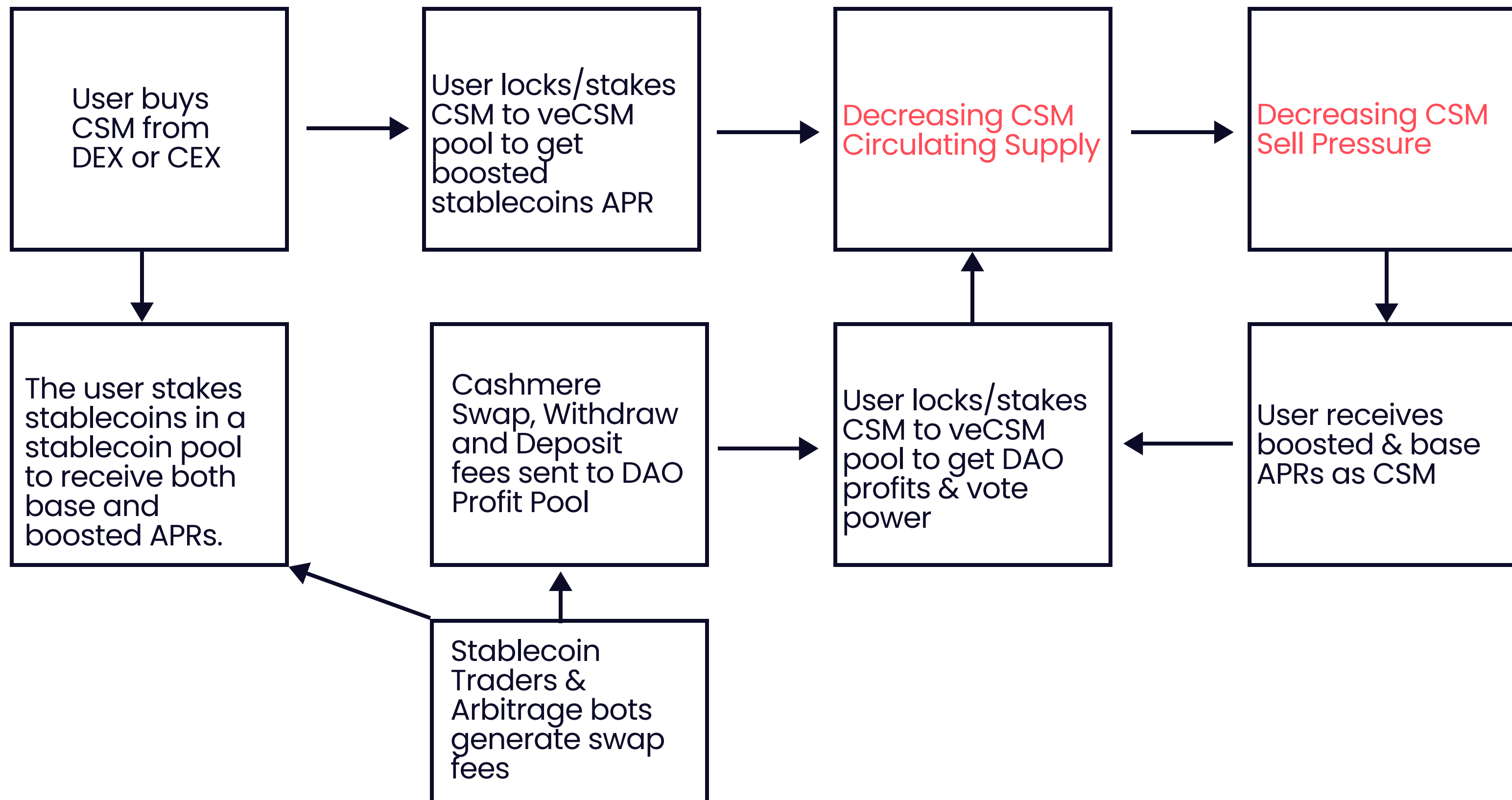
# Interest Rate Model

**TLDR:** The higher the compensation ratio, the greater the CSM emission is to the stablecoin account.



- Compensation Ratio = Asset / Liability

# Sustainability



**Powering the seamless  
stableswap & asset aggregator  
across 7 high value chains with  
just one integration**

# Cross-chain aggregator

Swap any asset between any chain with **best offers** !



Stable coins to be used



Assets to be used



Any asset that has  
liquidity on any DEX

# Stableswap Technical Overview

Stableswap

Aggregator

From

BALANCE 24689.905

DAI

Ethereum

24.689.905

MAX

To

BALANCE 24689.905

USDT

Polygon

24.500,400

MAX

Rate (After Fee)

1 UST = 1.017 USDT

Price Impact

0.05%

Fee

24.169.287 USDT

Minimum Received

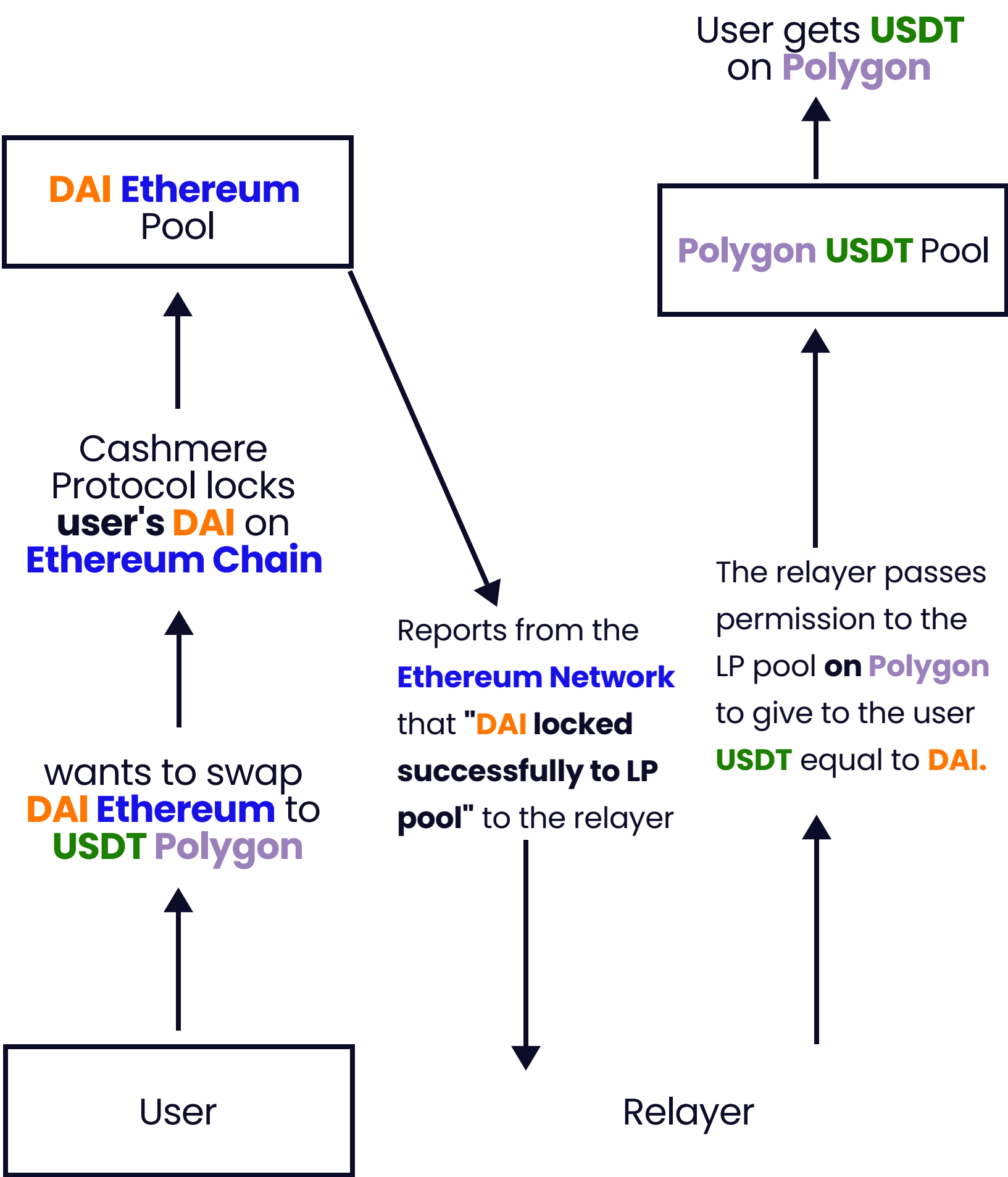
15.6235 USDT

Connect Wallet

DAI

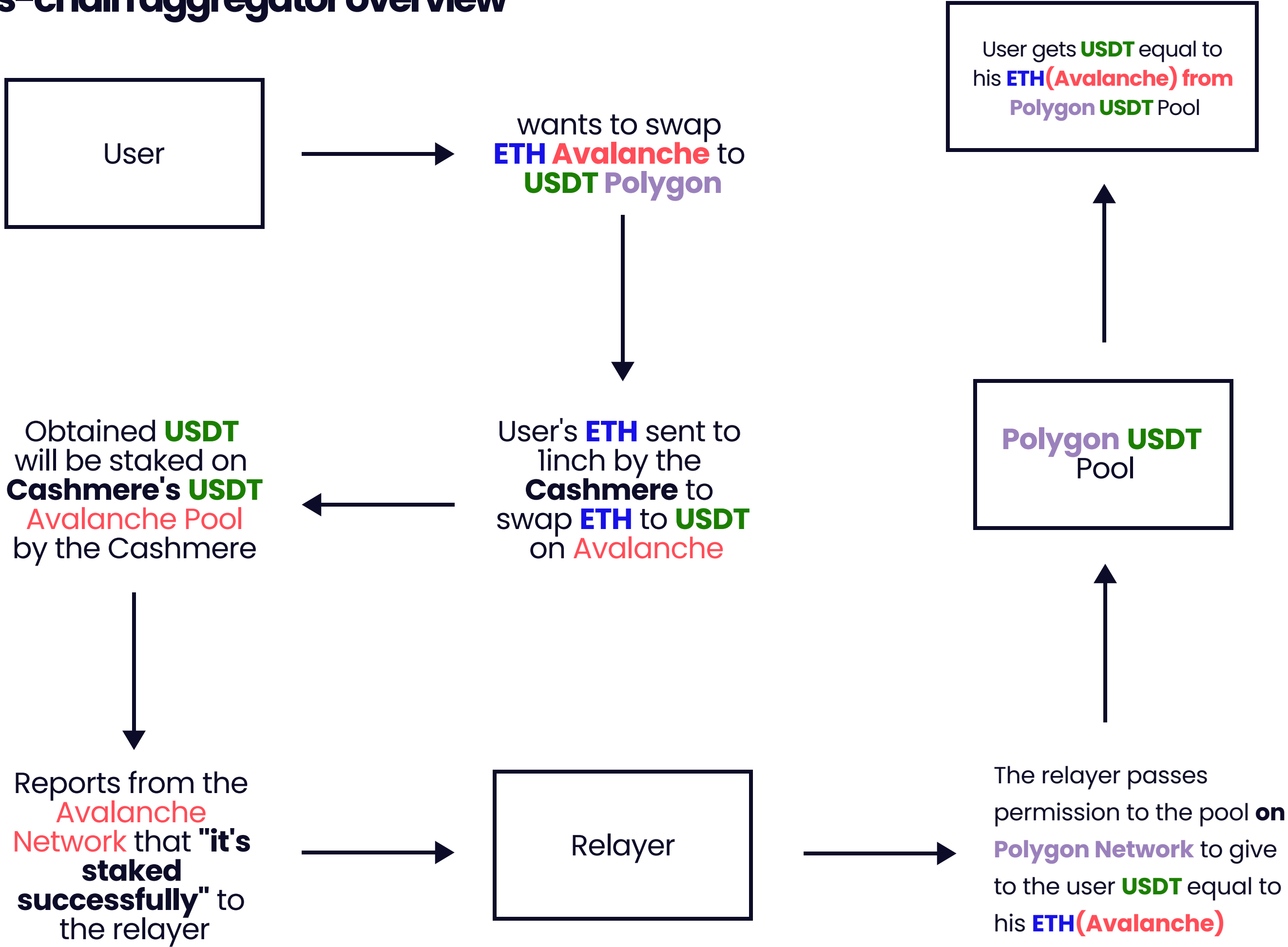
USDC

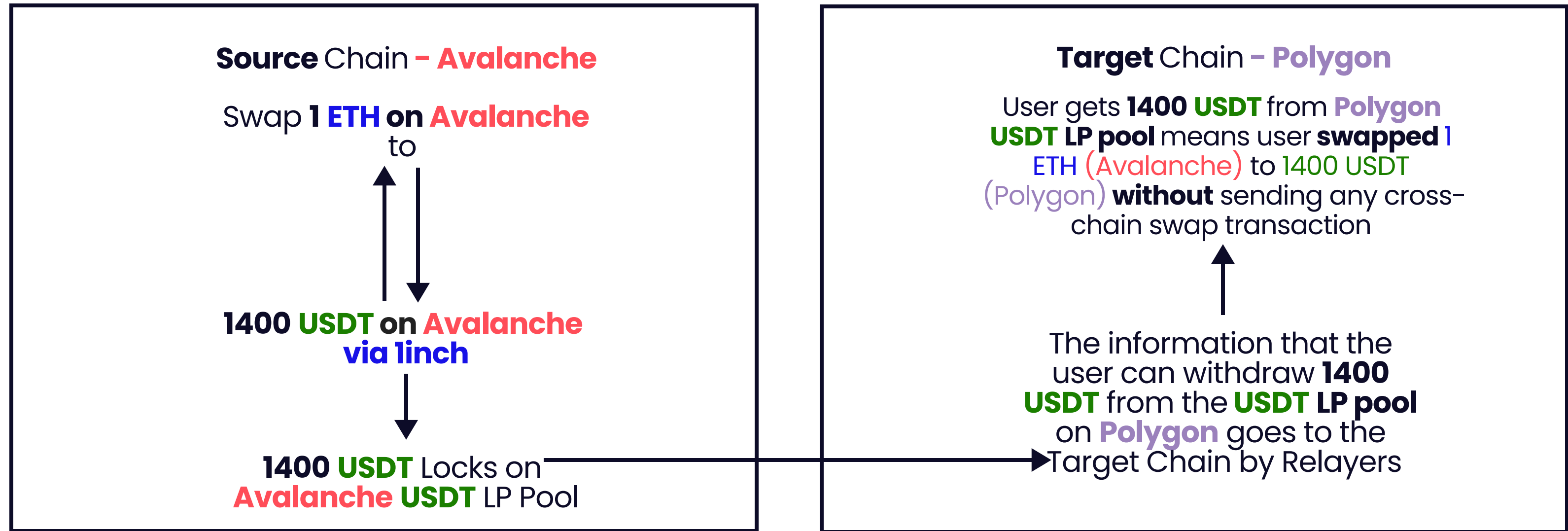
USDT





# Cross-chain aggregator overview

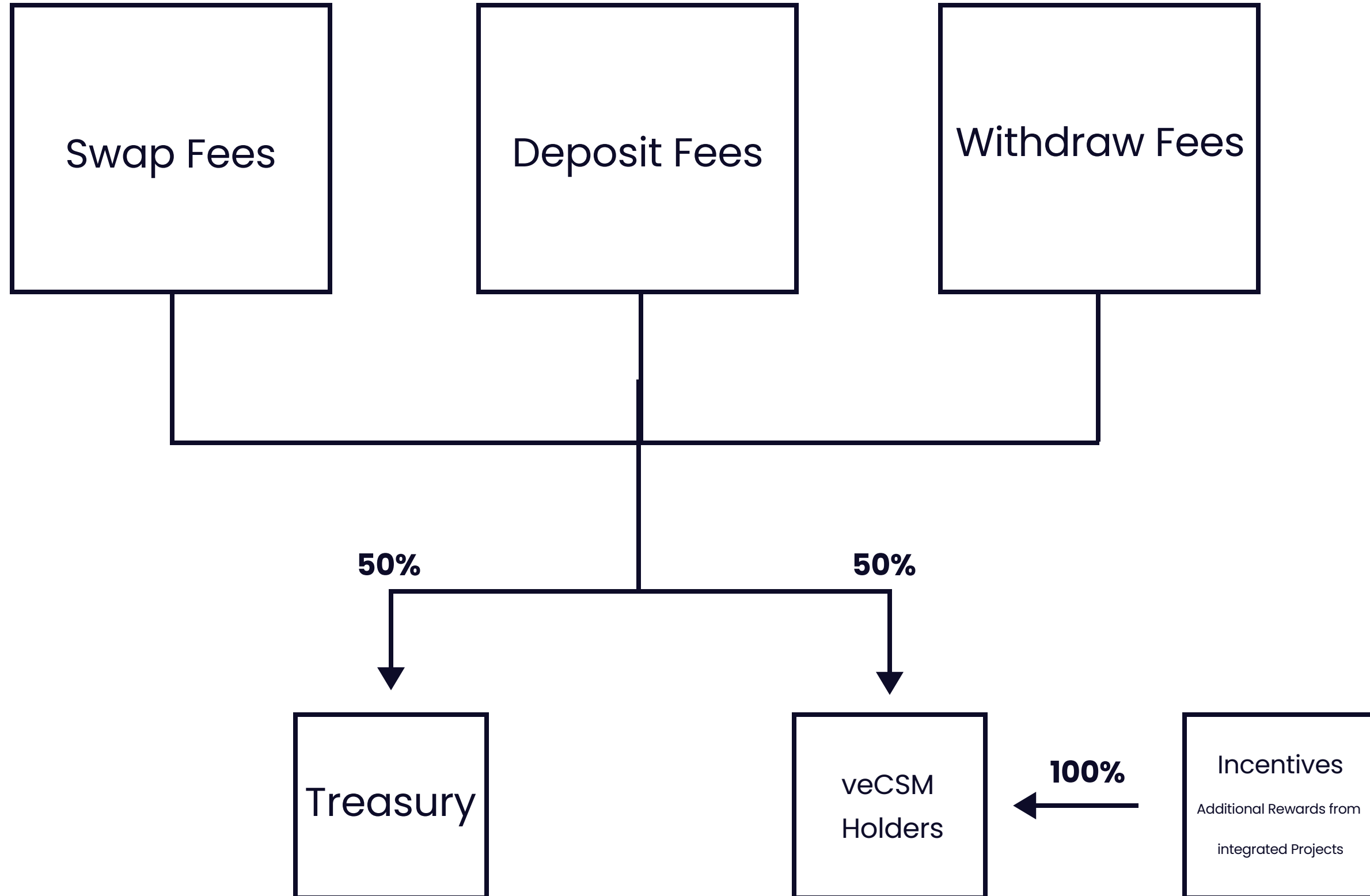





To show Cashmere's **MEV resistant** design;  
We consider a user scenario who wants to **swap** wrapped **ETH** on **Avalanche** Network to **USDT** on **Polygon** Network.

Users won't be sending a swap transaction between the chains. The swap of the assets will be occurring in each chain, **seperately**. Only the **information** of the amount of the asset that will be granted to users withdrawal from stable LP pool will be delivered between the chains. Therefore, MEV bots won't be able to attack during the cross-chain messaging period. This system **does not** completely eliminates the risk of MEV. But it will provide the slippage amounts as a regular **decentralized swap operation in a native network**. The cross-chain operation does not create an additional MEV attack risk.

# Business Model



# Value Proposition

	 <b>Cashmere</b>
<b>Security</b>	It is enough for the user to hold 1 native stablecoin that he trusts. <b>No bridge required.</b>
<b>Return of Investment</b>	Has DAO <b>gauge</b> and <b>lock</b> system, Cashmere provides completely <b>decentralized</b> and <b>fair reward</b> distribution.
<b>Slippage</b>	Cashmere <b>has the lowest slippage</b> with single-sided liquidity.
<b>Swap to</b>	<b>Any asset &amp; Any chain</b>
<b>Cross-chain</b>	Switching from any chain to other chain with interoperability <b>messaging protocol</b>

Project	Fully Diluted Marketcap	30 Days Total Revenue	P/S Ratio (yearly) FDV/Revenue
Uniswap	\$5,327,200,199	\$61M	6.36
Dydx	\$1,795,124,574	\$19M	6.07
Pancakeswap	\$2,363,808,268	\$13M	3.82
Balancer	\$468,059,187	\$2.8M	14.67
Curve	\$3,233,617,769	\$6M	35.76
Sushiswap	\$282,161,712	\$6M	3.30
Cashmere	\$65,000,000 (initial FDV)	\$750K – 1.5M (imaginated by similar dapps and liquidity)	3.61 – 7.22

# Security Measures

## 1-Price Oracle

- Cashmere tracks each token's exchange rate via Chainlink. If there is a stablecoin unpeg of more than **+2%** (max price deviation), swaps will be **stopped**.
- Withdrawals will **continue**.

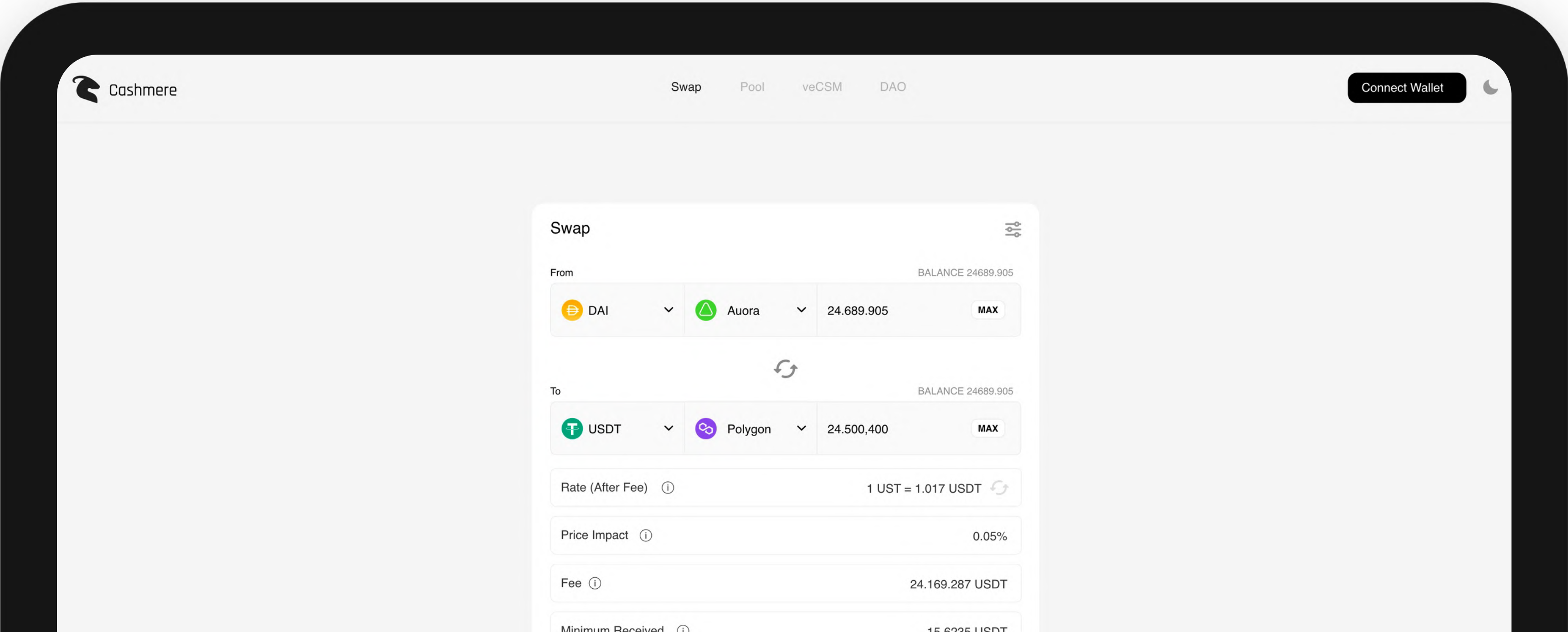
## 2-MEV Resistance

- The swap of the assets will be occurring in each chain, **seperately**.
- Only the **information** of the amounts of the assets will be transferred for users withdrawals will be delivered between the chains. **Not using the bridges**, which is vulnerable to attacks.
- Security measures in case of the MEV bot attacks during the cross-chain messaging period.

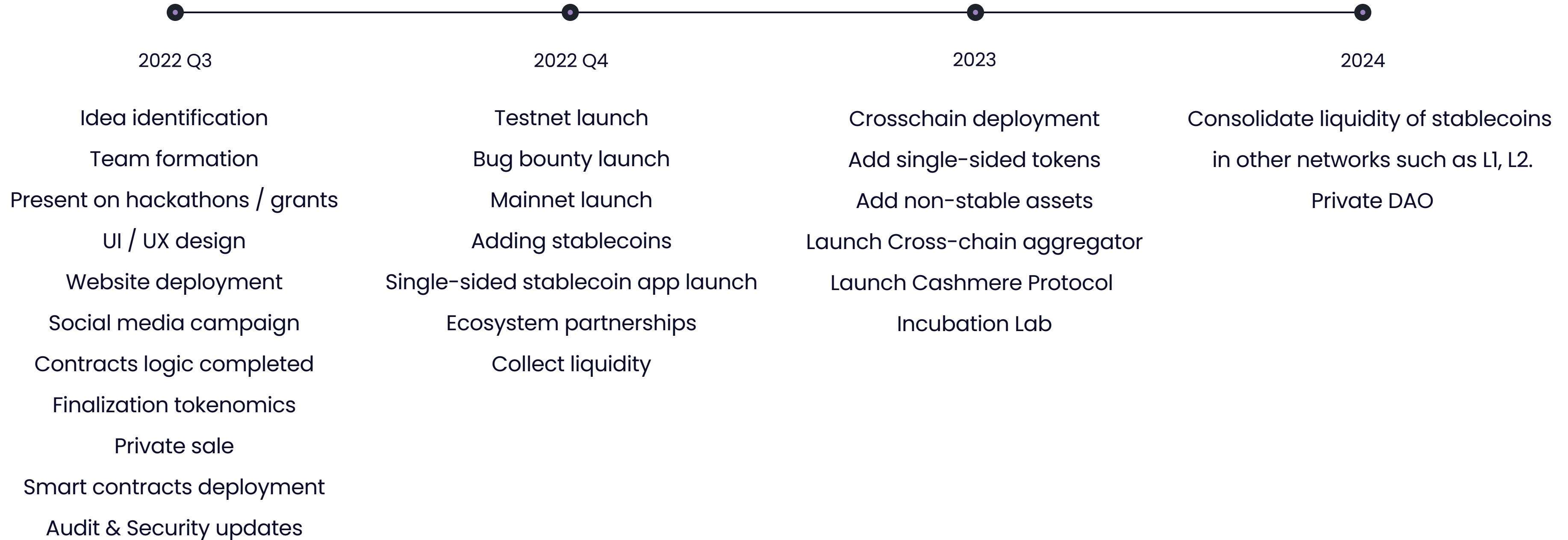
## 3-Upgradeability

- Cashmere Protocol is implemented as a set of **upgradeable smart contracts**.
- At early stages of the project, ProxyAdmin will be owned by Multisig.
- Later on, ProxyAdmin contract will be owned by Governance. It will be voted by veCSM holders.

# Cashmere Protocol UI/UX Alpha



# Roadmap



# Tokenomics – General

- Max supply: 100M CSM
- 50% LP Rewards – 50M

*Distribution depends on formula based emission rate.*

- 30% Treasury – 30M

*100% TGE – Used for launch liquidity and other community driven activities.*

- 10% Team & Core Contributors – 10M

*1-year full lockup, 36 months linear unlock thereafter.*

- 9% Investors – 9M

*– 2.25M Seed*

*– 6.75M Serie A*

*50% locked on voting escrow for 4 years, left-over 1 year full-lockup, 36 months linear unlock thereafter.*

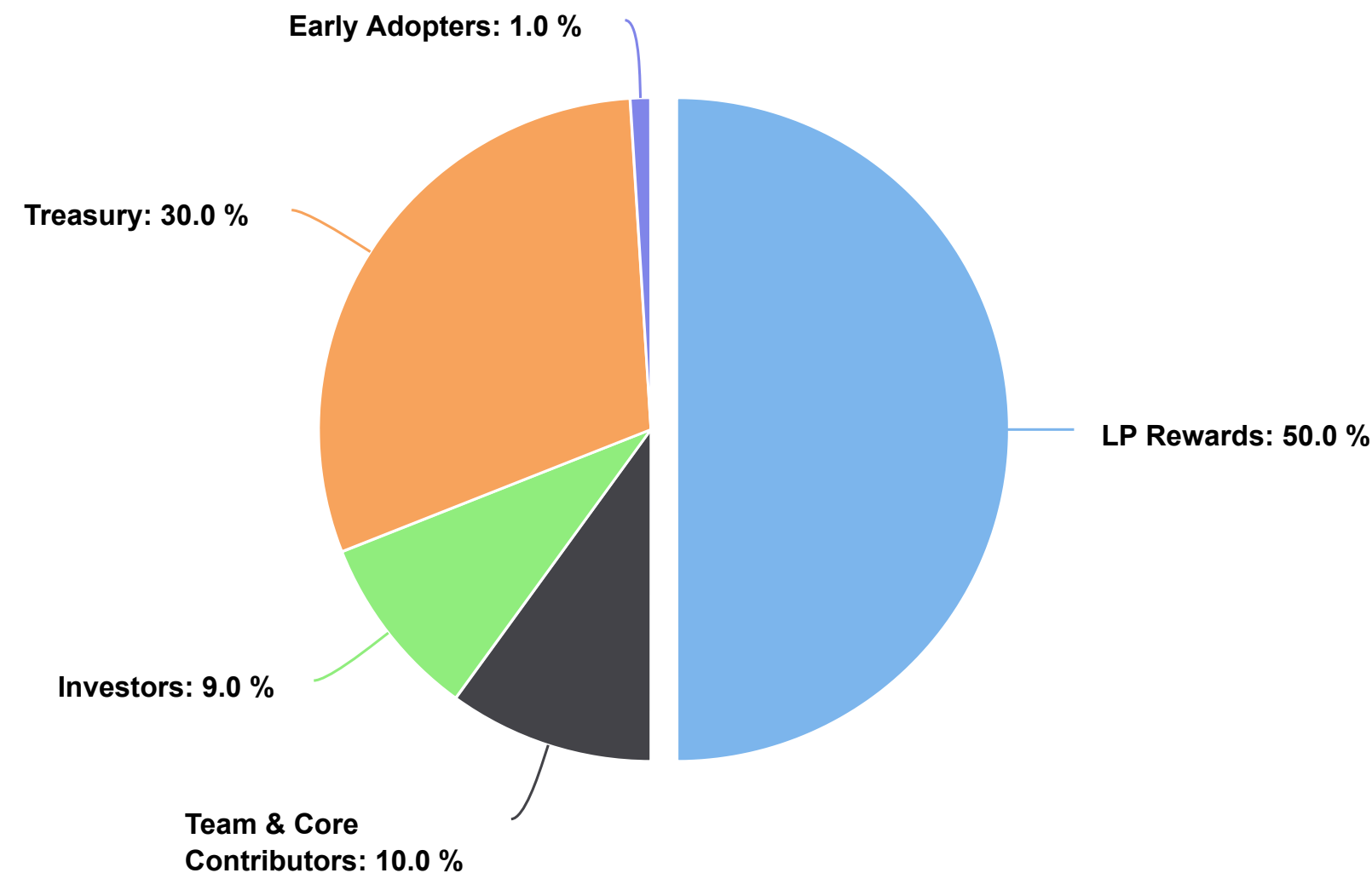
- 1% Early Adopters – 1M

*1-year full lockup, 12 months linear unlock thereafter.*

## Metrics

- *Valuation: \$52M*
- *Initial Circulating Supply without Liquidity: \$0*
- *Initial Circulating Supply with Liquidity: \$19.5M*
- *Listing Price: \$0.65*

Cashmere Token Distribution

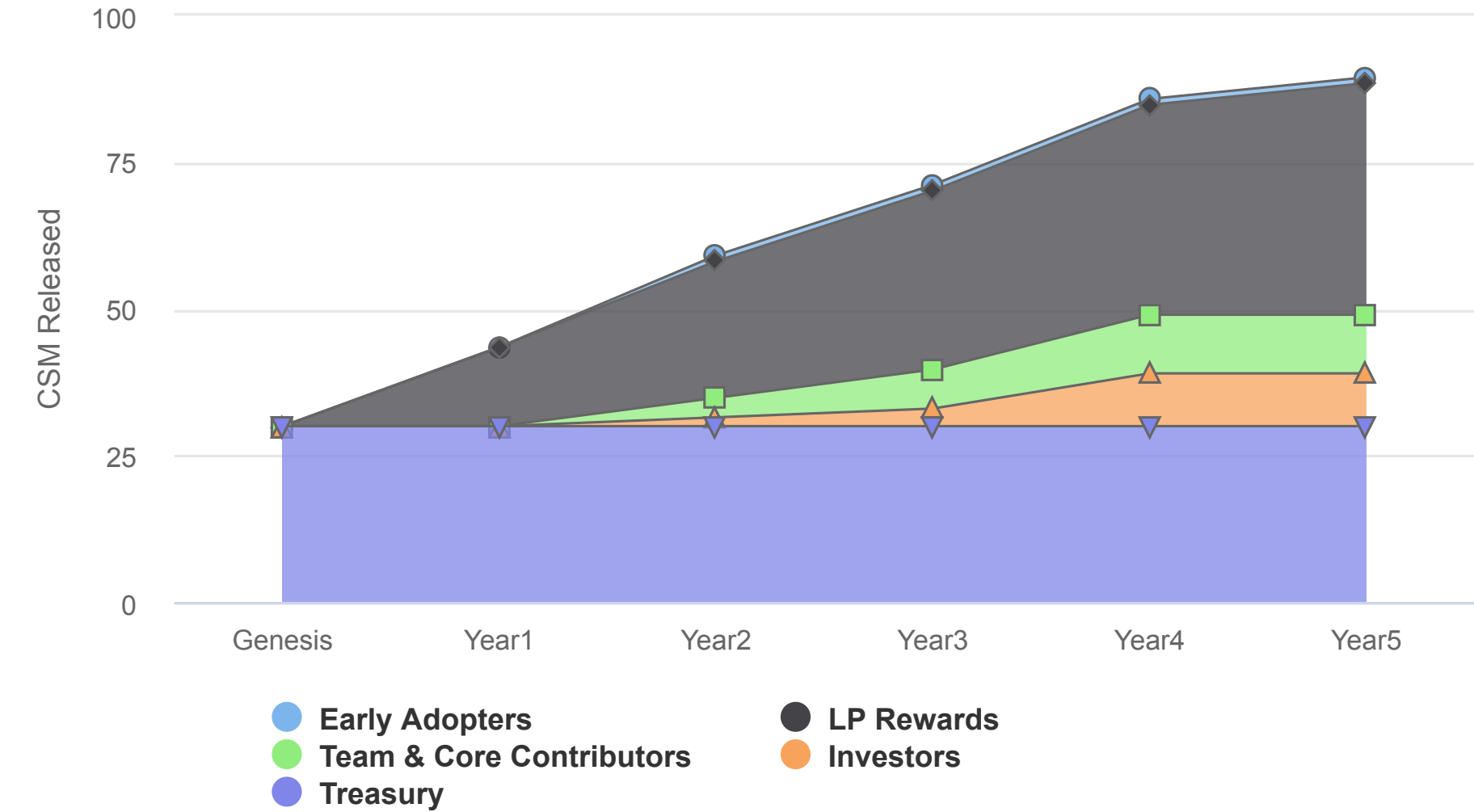




# Tokenomics – Inflation Rates

CSM Release Schedule

Source: docs.cashmere.exchange



Highcharts.com

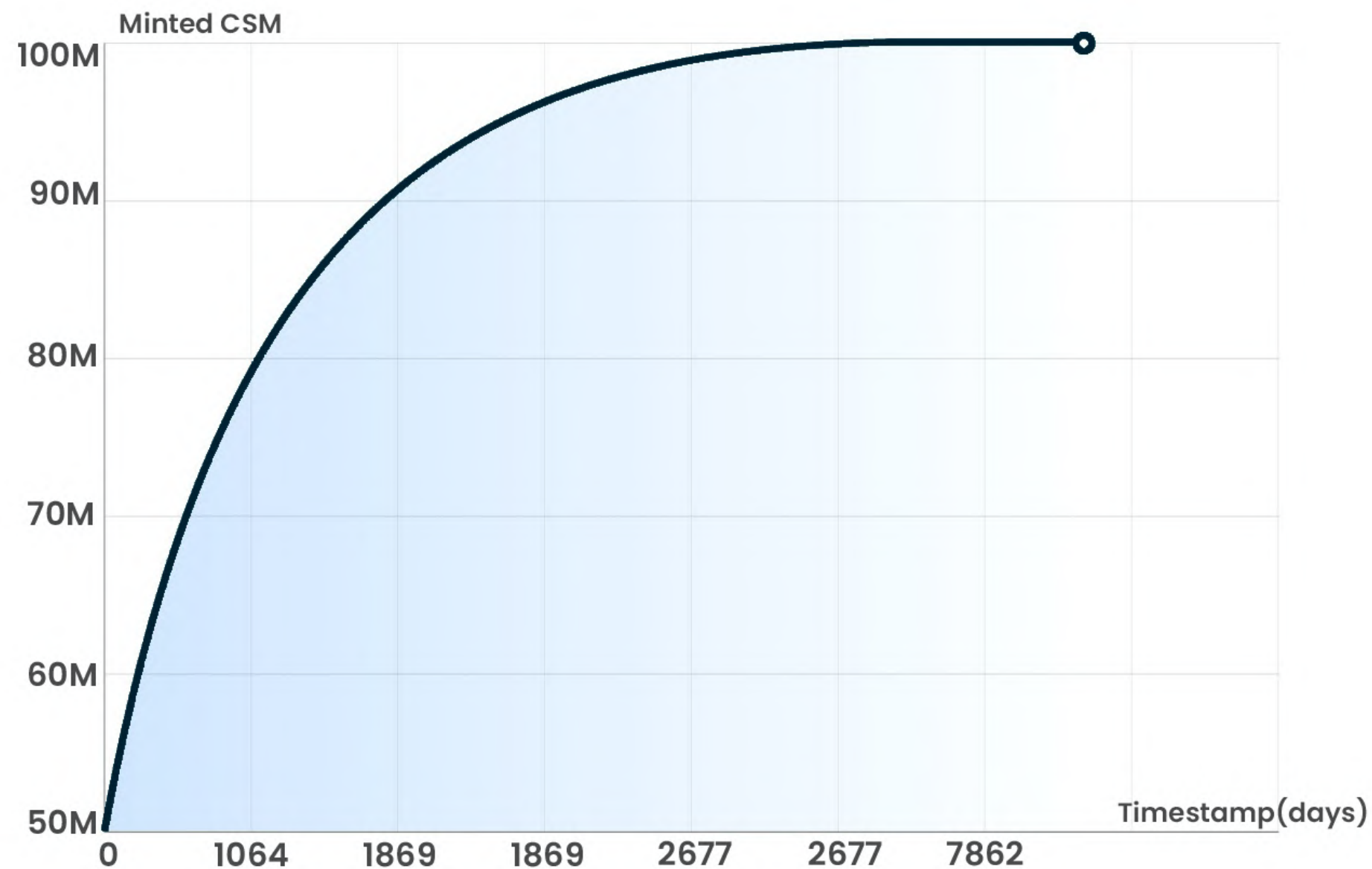
	Genesis	Year 1	Year 2	Year 3	Year 4	Year 5
LP Rewards	0	13,500,000	23,300,000	30,500,000	35,800,000	39,500,000
Treasury	30,000,000	30,000,000	30,000,000	30,000,000	30,000,000	30,000,000
Team & Core Contributors	0	-	3,333,000	6,666,000	10,000,000	10,000,000
Investors	0	-	1,500,000	3,000,000	9,000,000	9,000,000
Early Adopters	0	-	1,000,000	1,000,000	1,000,000	1,000,000
Total Unlocked Tokens	30,000,000	43,500,000	59,133,000	71,166,000	85,800,000	89,500,000
CSM_Inflation (%)	-	45.0	35.9	20.3	20.6	4.3

# CSM Token – Emission Rates



## CSM Minted per Block

Ratio reduces over time until all CSM is minted



- Supply stablecoin LPs and lock CSM to earn CSM and platform fees.
- CSM will be minted depends on emission formula.
- Minted CSM / Timestamp mint ratio reduces every 100,000 CSM.
- **CSM Mint Rate Formula:**

```

1 //constants
2 let cliffSize = 100000 * 1e18; //new cliff every 100,000 tokens
3 let cliffCount = 1000; // 1,000 cliffs
4 let maxSupply = 100000000 * 1e18; //100 mil max supply
5
6 //first get total supply
7 let csmTotalSupply = await csm.totalSupply();
8
9 //get current cliff
10 let currentCliff = csmTotalSupply / cliffSize;
11
12 //if current cliff is under the max
13 if(currentCliff < cliffCount ){
14     //get remaining cliffs
15     let remaining = cliffCount - currentCliff;
16
17     //ratio of remaining cliffs to total cliffs will be CSM Rate
18     CSM Rate = remaining cliffs / cliffCount;
19
20
21
22     return 0;
23 }
  
```

**Expect nothing  
less than  
the best  
experience.**

