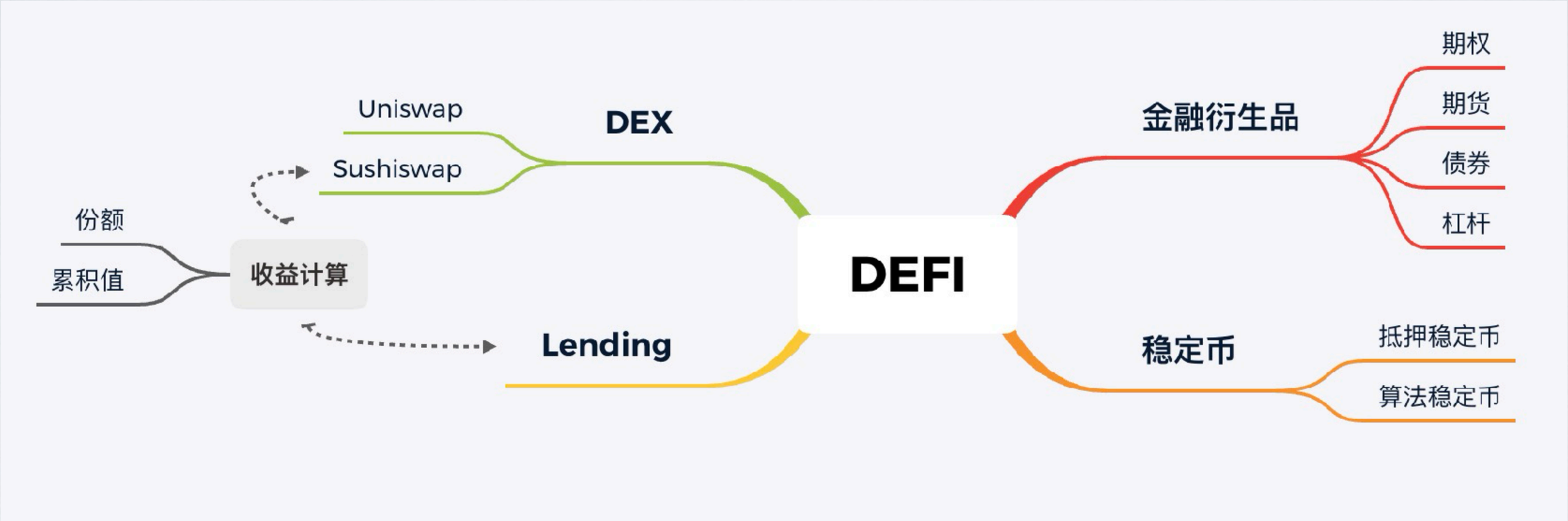

DEFI
DEX
Uniswap
Sushiswap
收益计算
份额
累积值
Lending
金融衍生品
期权
期货
债券
杠杆
稳定币
抵押稳定币
算法稳定币

继续深入 DEX

Tiny 熊

要点

- 如何兑换 ETH
- 什么是无常损失
- 什么是闪电兑换
- 如何获取价格、价格预言机 - Chainlink
- Uniswap 的后续版本迭代 （V3、V4， Uniswap X）
- 其他的 DEX： SushiSwap、PancakeSwap
- 交易聚合器

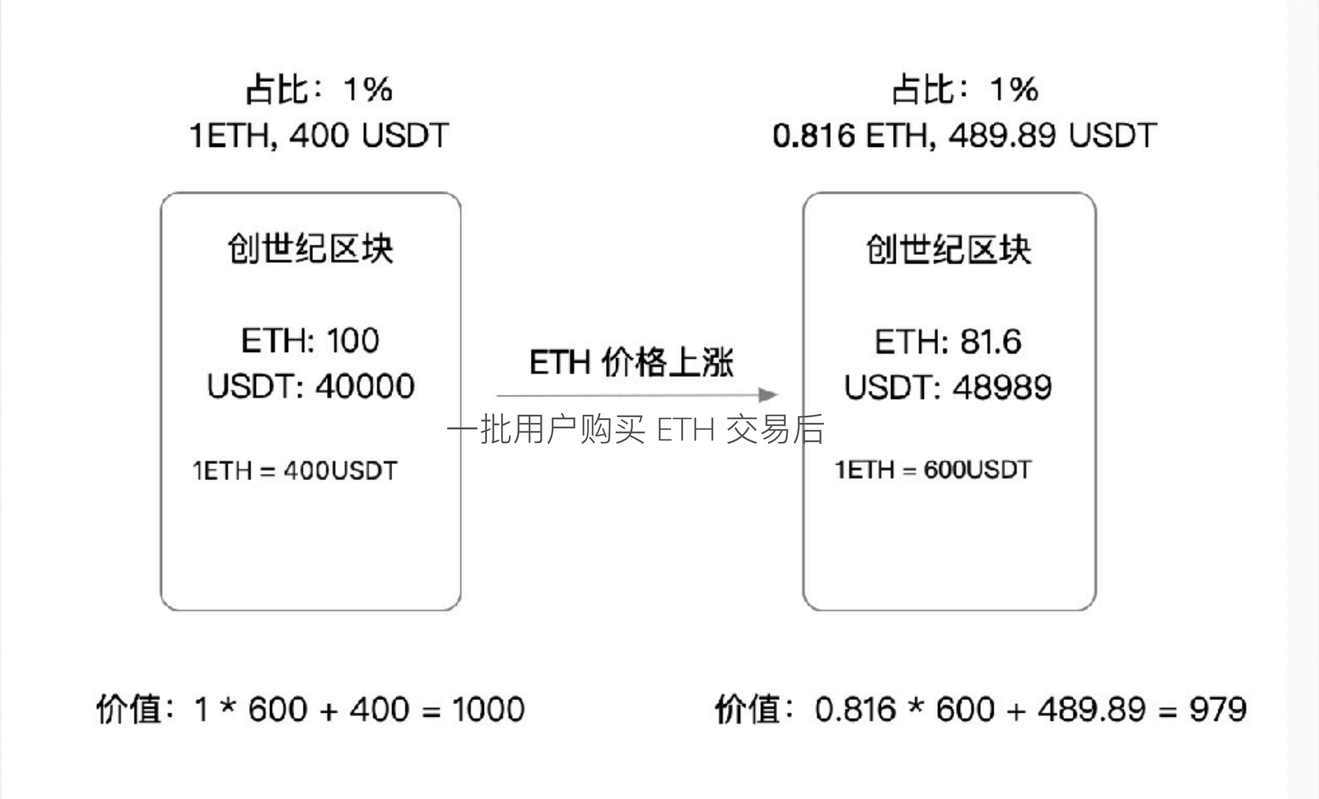
兑换 ETH – WETH

- WETH 是 Ether 的 ERC20 包装(Wrap-ETH)， 以便方便的和 ERC20 对接
- 1 WETH = 1 ETH
- WETH 合约（WETH9）
  - https://etherscan.io/token/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2#code
  - deposit()
  - withdraw()
- 在兑换(卖)或添加流动性时，先调用一次 WETH.deposit()

https://github.com/Uniswap/v2-periphery/blob/master/contracts/UniswapV2Router02.sol

Uniswap – 无常损失

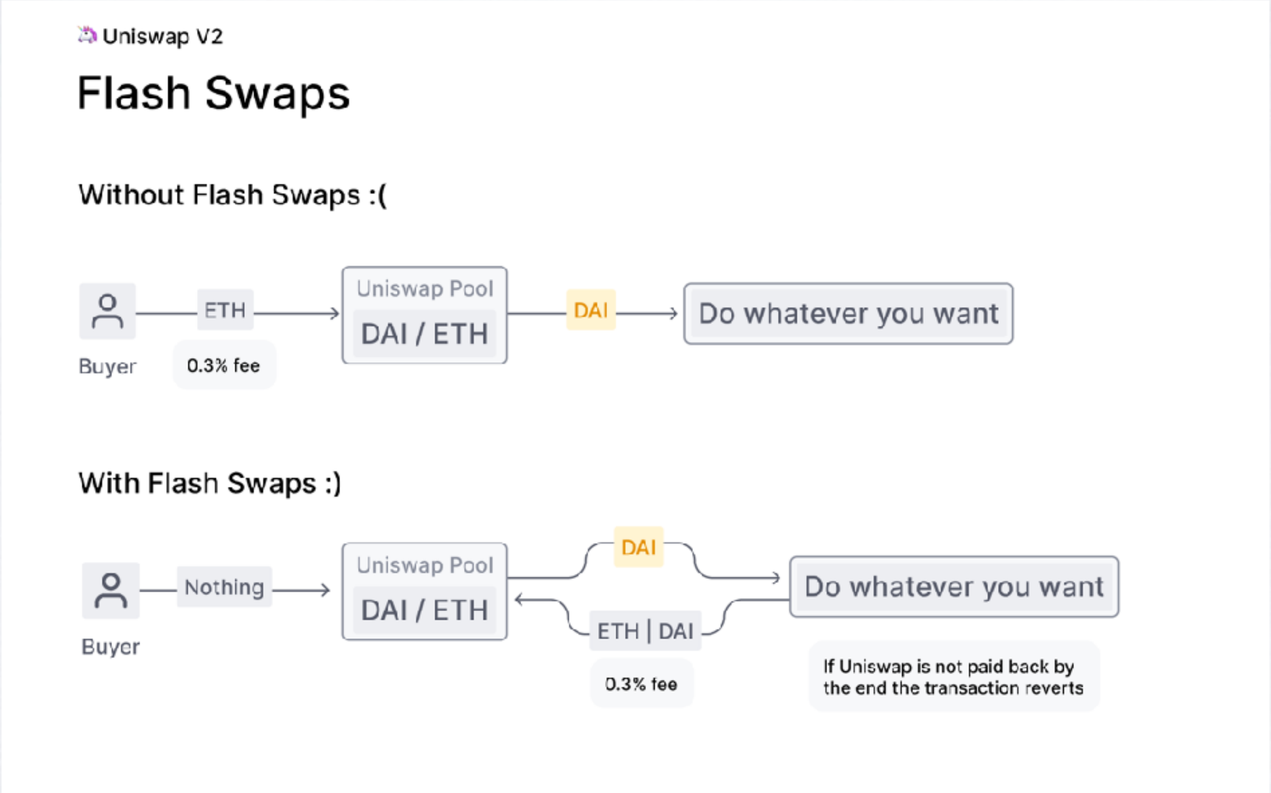
- 流动性提供者无常损失：一对代币存入Uniswap后，如果一种代币以另一种进行计价的价格上升，在价格上升后取出，总价格比原价值低一些，低的部分就是损失。
- 思考：为什么会造成损失?



https://zhuanlan.zhihu.com/p/268435169

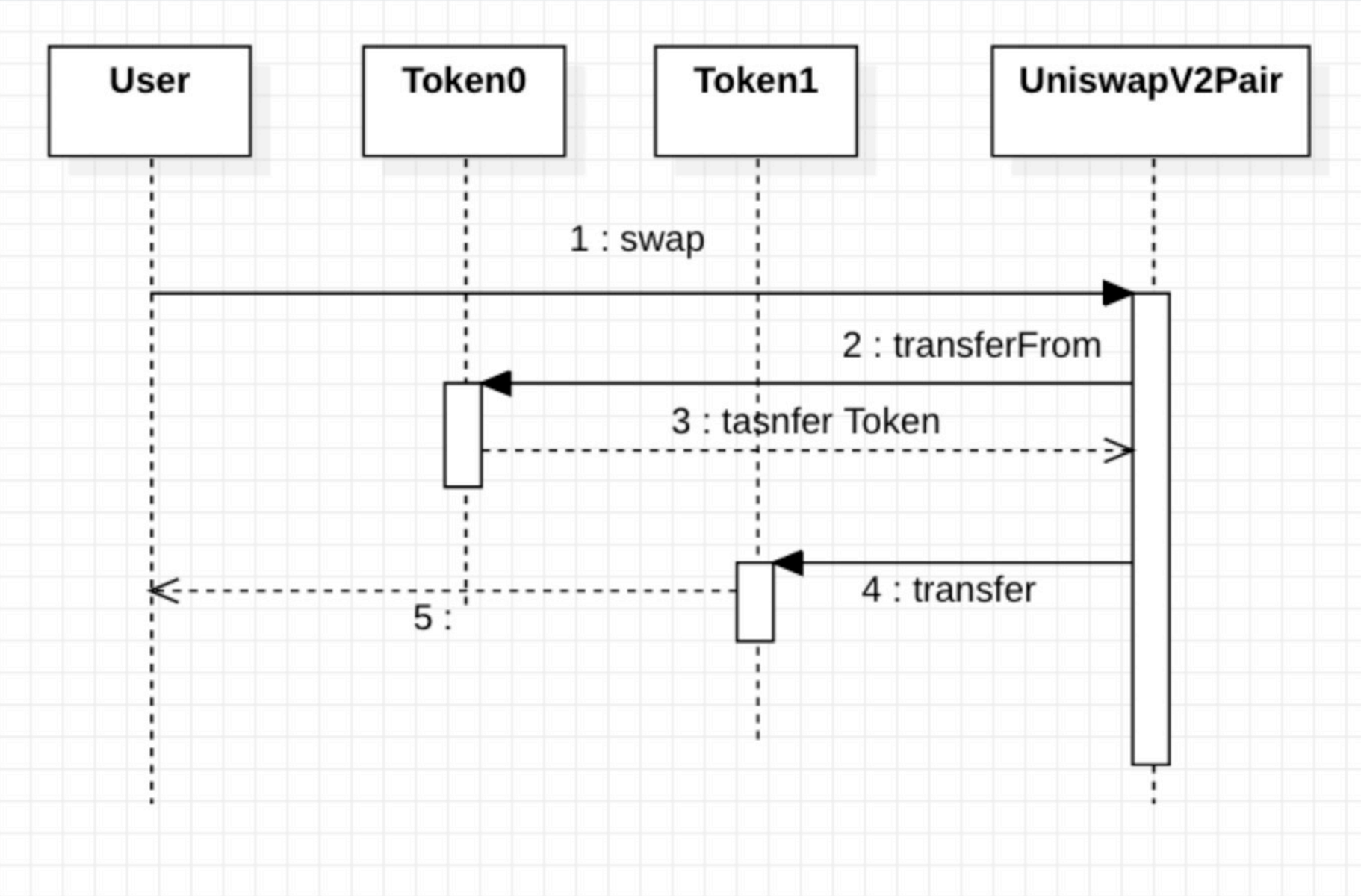
Uniswap 闪电兑

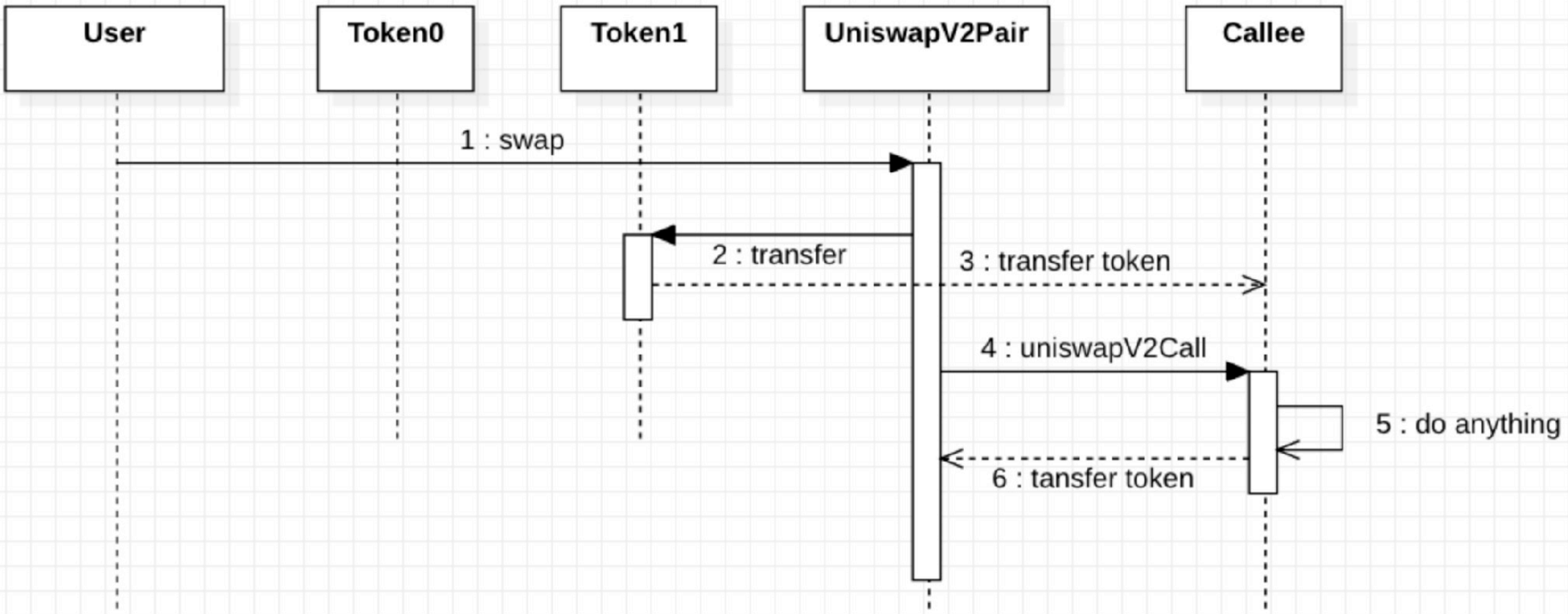
- 先从流动性池中“借出”任意数量的代币， 只要在交易结束前（同一个交易中）将它们归还或偿付等值资产即可。



uniswapV2Callhook 函数

普通兑换


User
Token0
Token1
UniswapV2Pair
1 : swap
2 : transferFrom
3 : tasnfer Token
4 : transfer
5 :

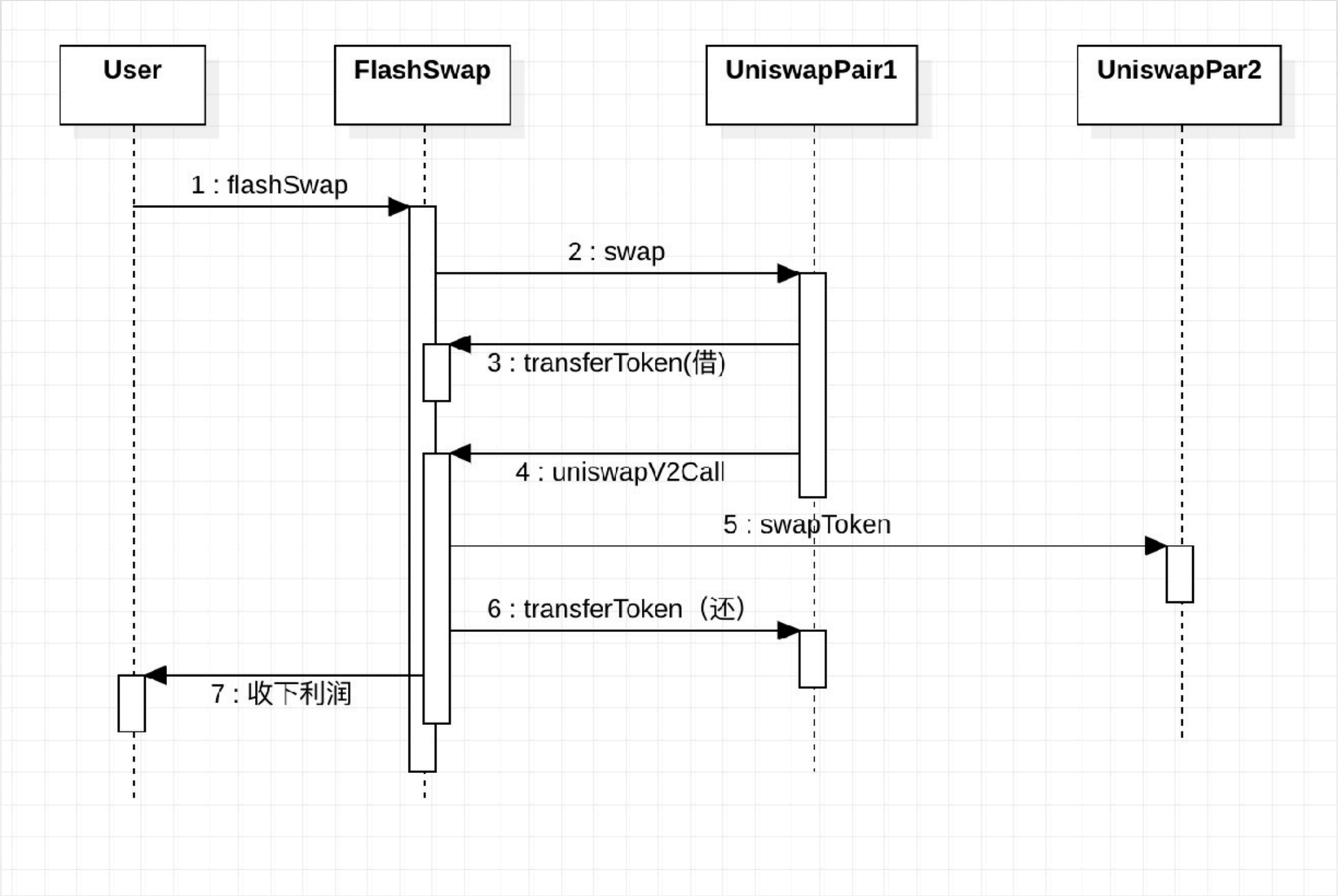

User
Token0
Token1
UniswapV2Pair
Callee
1 : swap
2 : transfer
3 : transfer token
4 : uniswapV2Call
5 : do anything
6 : tansfer token

闪电兑换

Uniswap 闪电兑换套利

- 假设有两个 DEX 有价差：
  - 在 Pair1 中 1 TokenA = 2 TokenB
  - 在 Pair2 中 1.5 TokenA = 2 TokenB
- 如何在这两个 DEX 中套利?
  - 我们可以从 Pair1 中借出 2 TokenB， 在Pair2中兑换处 1.5 TokenA， 把 1 TokenA 还回 Pair1
  - 净赚 0.5 TokenA

代码库： https://github.com/xilibi2003/v2-periphery
文件： contracts/FlashSwap.sol



代码库： https://github.com/xilibi2003/v2-periphery

文件： contracts/FlashSwap.sol

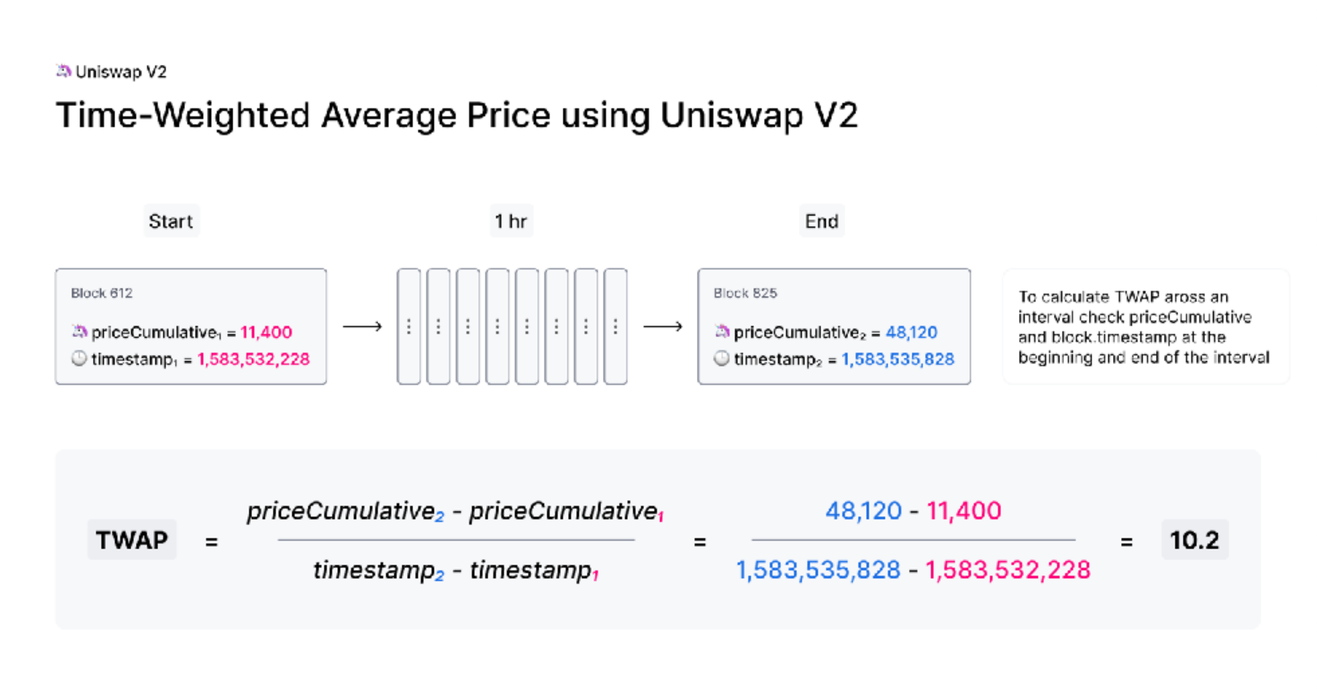
获取某个 Token 价格

- 很多情况，需要在链上获取代币价格，例如计算抵押品（存款）的价值等。
- 要注意：即时价格 spot price（用 getAmountsOut ）是不安全的，为什么？
  - 当流动性低时，价格波动很大，也容易在一个交易中被操作（尤其是使用闪电兑/贷)
- 应该使用时间加权价格 TWAP（一段时间内的价格），防止价格操纵

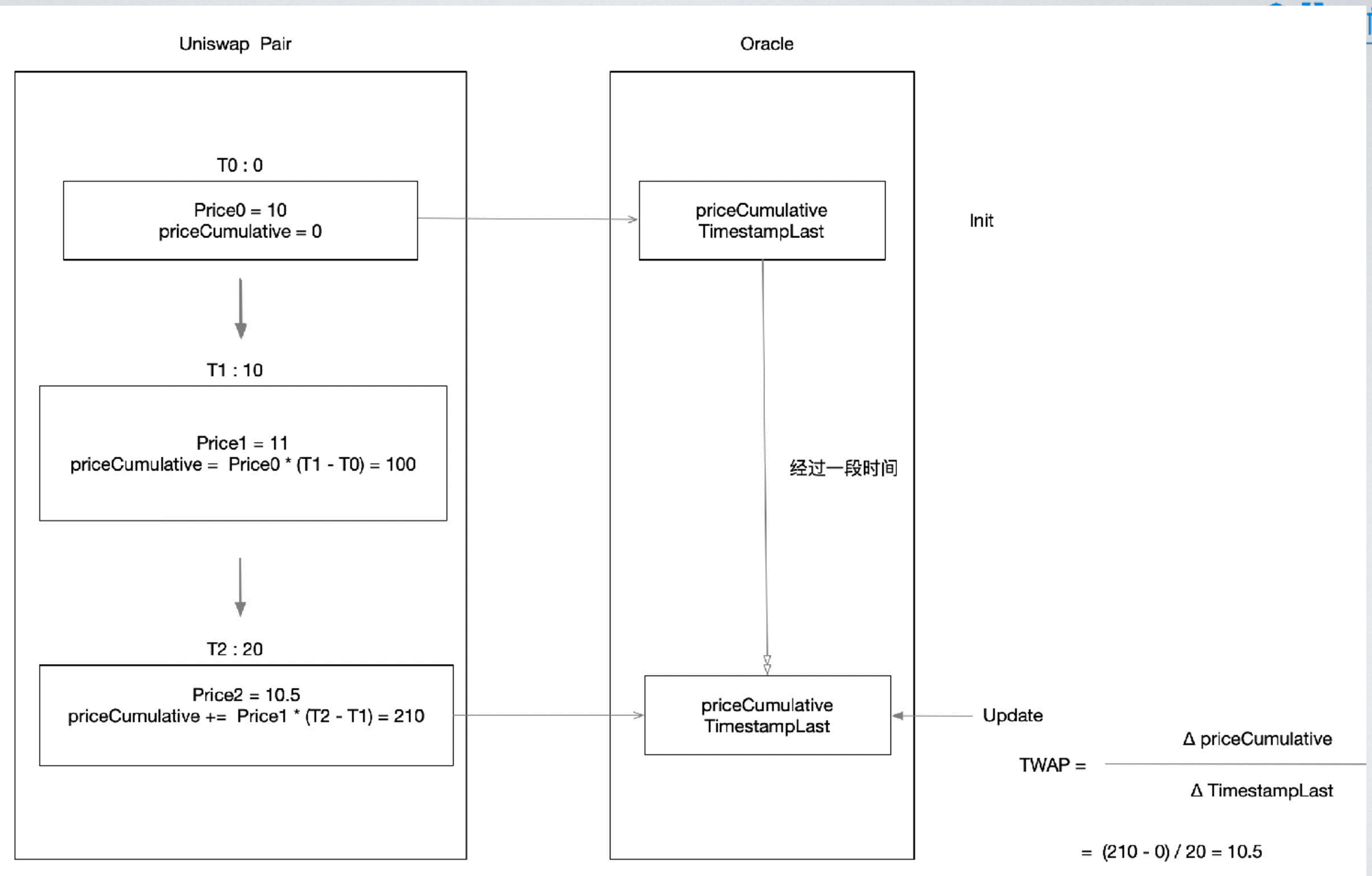
https://github.com/Uniswap/v2-periphery/blob/master/contracts/UniswapV2Router02.sol

Uniswap TWAP 价格

- TWAP = Time-Weighted Average Price，即时间加权平均价格
- 在 Uniswap V2 Core 里，每次交易都会更新：price0CumulativeLast、price1CumulativeLast 他们记录了 token 的累加价格
  - priceCumulativeLast1 = priceCumulativeLast1 + price2 * timeElapsed;（当前价格累计值 = 上一次价格累计值 + 当前最新价 * 持续时间）
  - blockTimestampLast 记录了上次的时间



TWAP 价格演算


Uniswap Pair
Oracle
T0 : 0
Price0 = 10
priceCumulative = 0
priceCumulative
TimestampLast
Init
T1 : 10
Price1 = 11
priceCumulative = Price0 * (T1 - T0) = 100
经过一段时间
T2 : 20
Price2 = 10.5
priceCumulative += Price1 * (T2 - T1) = 210
priceCumulative
TimestampLast
Update
TWAP = Δ priceCumulative / Δ TimestampLast
= (210 - 0) / 20 = 10.5

如何接入 TWAP 价格

- 在我们的协议合约里间隔调用 update() 记录累计价格并计算平均价格

```
function update() external {
    (uint price0Cumulative, uint price1Cumulative, uint32 blockTimestamp) =
        UniswapV2OracleLibrary.currentCumulativePrices(address(pair));
    uint32 timeElapsed = blockTimestamp - blockTimestampLast;

    require(timeElapsed >= PERIOD, 'ExampleOracleSimple: PERIOD_NOT_ELAPSED');

    price0Average = (price0Cumulative - price0CumulativeLast) / timeElapsed;
    price1Average =  price1Cumulative - price1CumulativeLast) / timeElapsed ;

    price0CumulativeLast = price0Cumulative;
    price1CumulativeLast = price1Cumulative;
    blockTimestampLast = blockTimestamp;
}
```

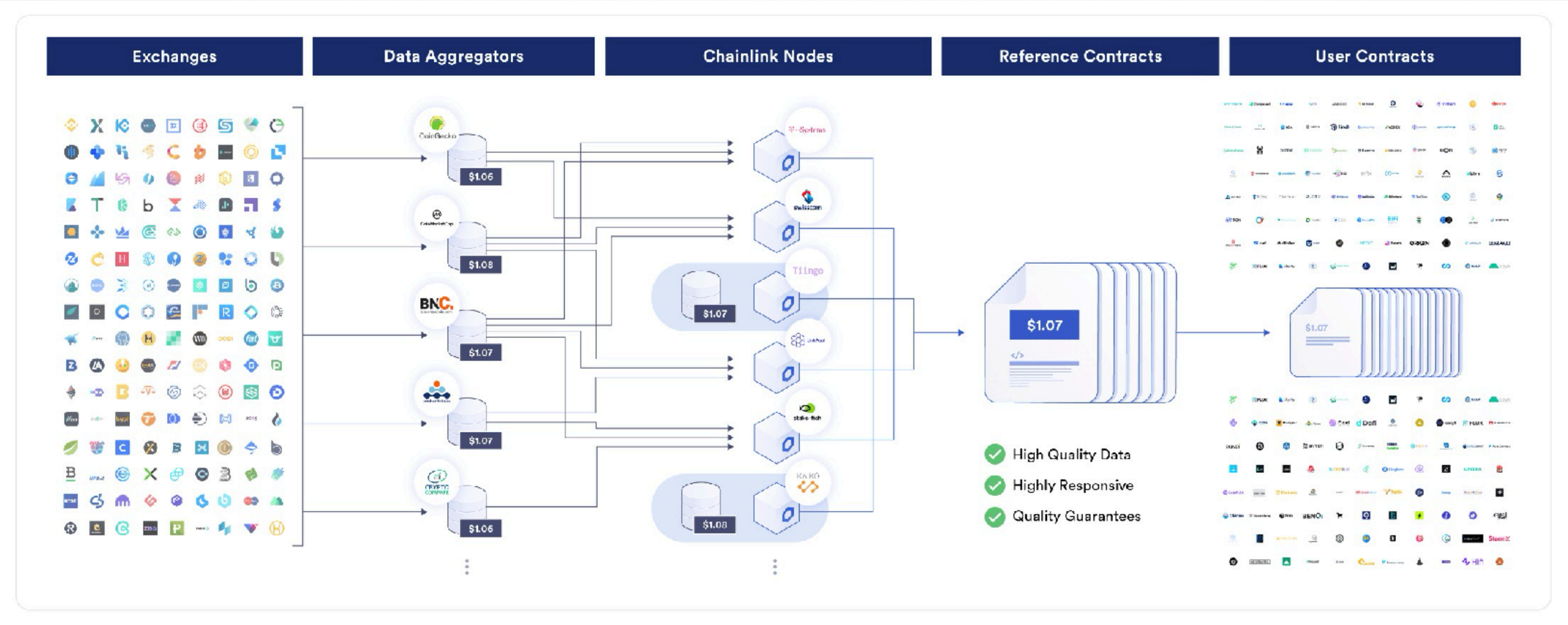
代码：https://github.com/Uniswap/v2-periphery/blob/master/contracts/examples/ExampleOracleSimple.sol

预言机（Oracle）服务

- 自己不断获取TWAP 价格的成本是较高的，有时也可以使用预言机服务
- 预言机应为链上合约无法获取外部的价格，预言机作为一个链上、链下的桥梁、把链下的数据转递到链上。
- 典型的Oracle 服务商：
  - ChainLink、Pyth Network（Solana）、API3

ChainLink 价格预言机

- Data Feeds: 由一组分散的独立节点运营商从许多数据源汇总而来（通常），用户链接对应的Chainlink合约即可获取到价格。
- Data Feeds： 提供主流资产（数字资产及股票）的价格或指数
  - 也提供：NFT 地板价、贮备证明



获取预言机数据

- 预言机节点会将价格推送到 Aggregator 合约，因此我们只需要接入 Aggregator 。
- 在 Chainlink 中找到 Aggregator 合约地址，例如：https://data.chain.link/feeds/ethereum/mainnet/eth-usd

```
import "@chainlink/contracts/src/v0.6/interfaces/AggregatorV3Interface.sol";
contract PriceConsumerV3 {
    AggregatorV3Interface internal priceFeed;

    constructor() public {
        priceFeed = AggregatorV3Interface(feedAddress);
    }

    function getLatestPrice() public view returns (int) {
        (uint80 roundID, int price,
            uint startedAt, uint timeStamp,
            uint80 answeredInRound) = priceFeed.latestRoundData();
        return price;
    }
}
```

练习题

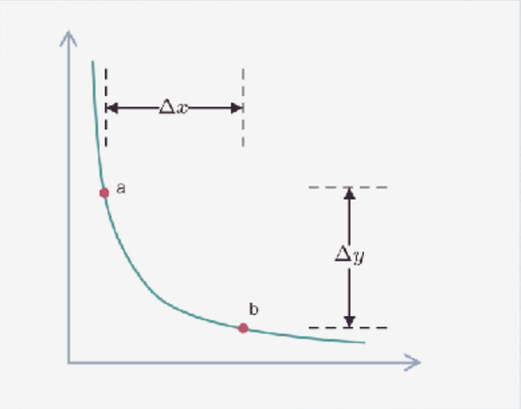
- 编写一个合约去获取 LaunchPad 发行的 Meme 的TWAP 价格， 请在测试中模拟不同时间的多个交易（兑换）。

https://decert.me/quests/ff20bbfe-0345-4f32-8ca3-fa77b3a0d6cb

Uniswap 协议的演进

V1 只支持与 ETH 对换，使用 vyper 开发

UNISWAP V2 不足



- 在所有价格段提供资金（0,infinity），资金利用率低， 滑点高
  - 资金利用率（Capital Efficiency） = 实际使用的资金/ 所提供的资金
  - 价格滑点（slippage）：一次交易使价格改变的程度

	x	y	
P = 1	1000	1000	
p=1.2	1100	909.1	
	100	-90.9	
价格波动 20%		资金利用率 ～9%	

要让价值平稳必须很高的流动性

UNISWAP V3

- 通过集中式流动性提高资金利用率

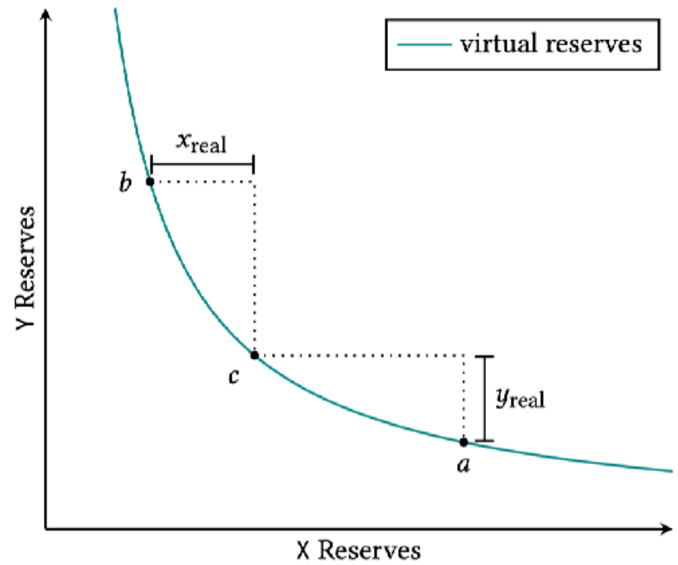


Figure 1: Simulation of Virtual Liquidity

1. 价格通常两点之间波动（a点和b点）
2. c 到 a， 消耗资金为 Yreal
3. c 到 b， 消耗资金为 Xreal
4. 若在 [a,b] 提供流动性，仅需提供Xreal Yreal

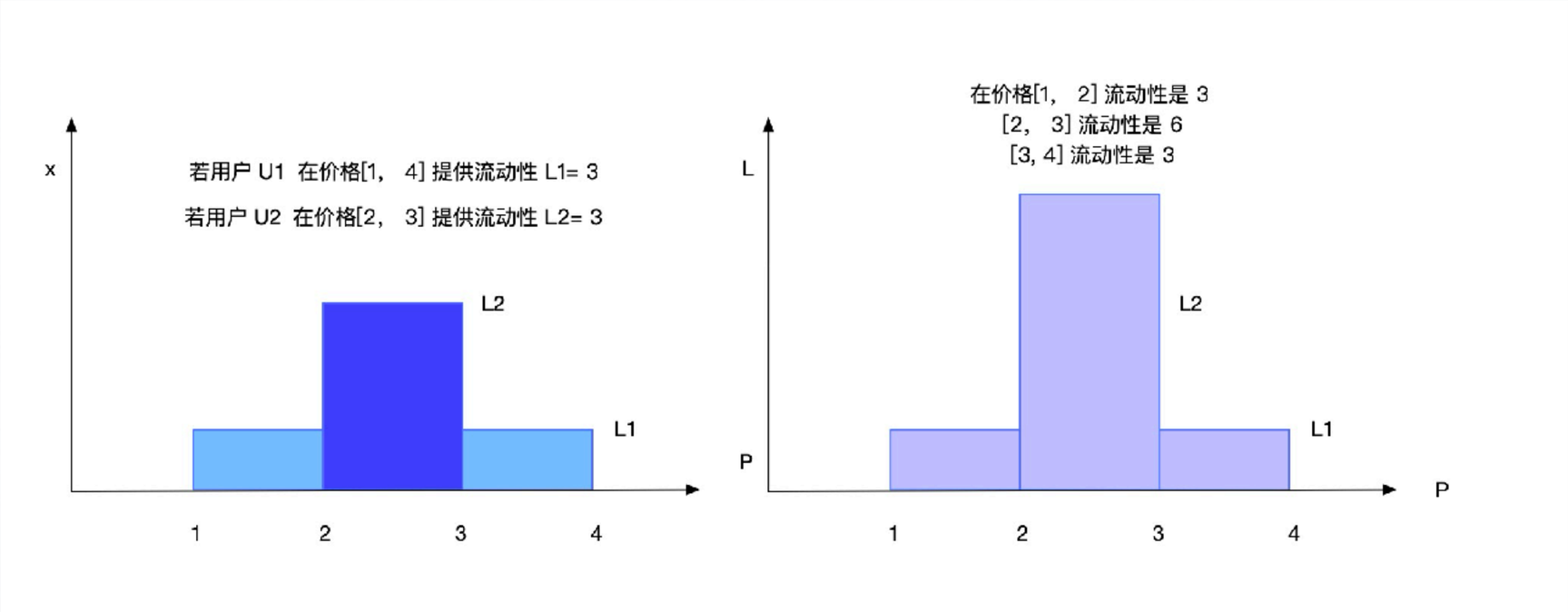
引入虚拟流动性 K = (X+Xvirtual) * (Y+Yvirtual)

UNISWAP V3

- 引入Tick，使用等比数列将价格区间分段，添加流动性只能选择对应的离散点作为价格边界
- 流动性聚合， 根据 tick 来聚合流动性

$$P_{\text{tick}} = 1.0001^{\text{tick}}$$

tick 之间的价格相差约 0.01%

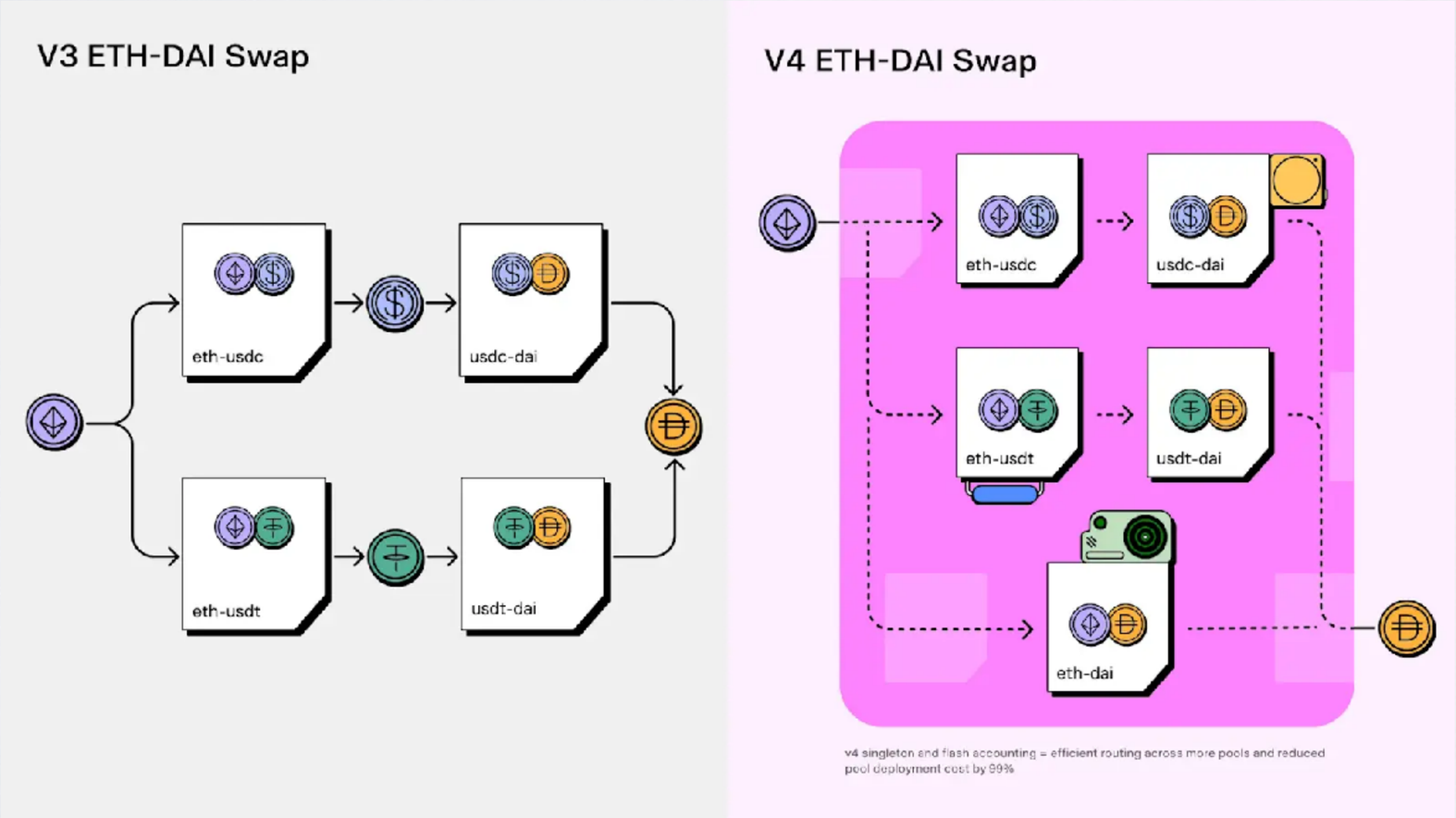


Uniswap V3 参考文章

- 白皮书：https://uniswap.org/whitepaper-v3.pdf
- 中文版白皮书：https://f00wcc7jp0.feishu.cn/file/boxcnKWqffRmWslUkik6jlfqZ7b
- Uniswap V3 文档：https://docs.uniswap.org/protocol/reference/smart-contracts
- Uniswap v3 分析：https://liaoph.com/tags/#Uniswap
- Uniswap v3 b 站：https://www.bilibili.com/video/BV1go4y1X7Kx
- 合约代码结构：https://j1mmy.fi/uniswapv3/pool
- 函数分析：https://www.desmos.com/calculator/oiv0rti0ss?lang=zh-CN

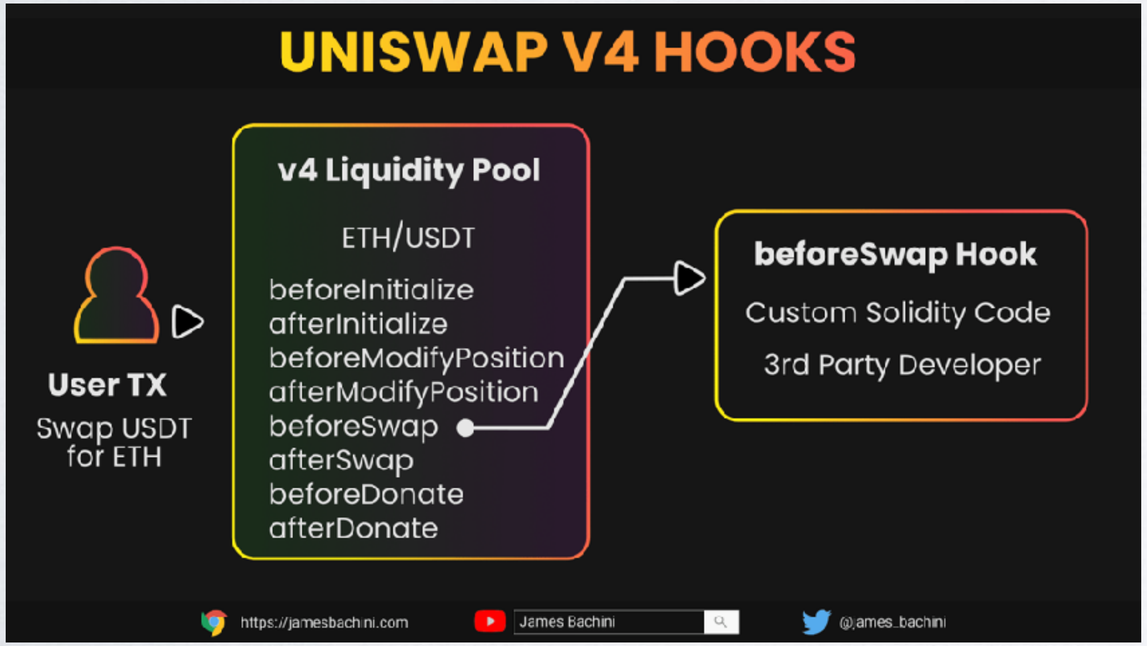
UNISWAP V4

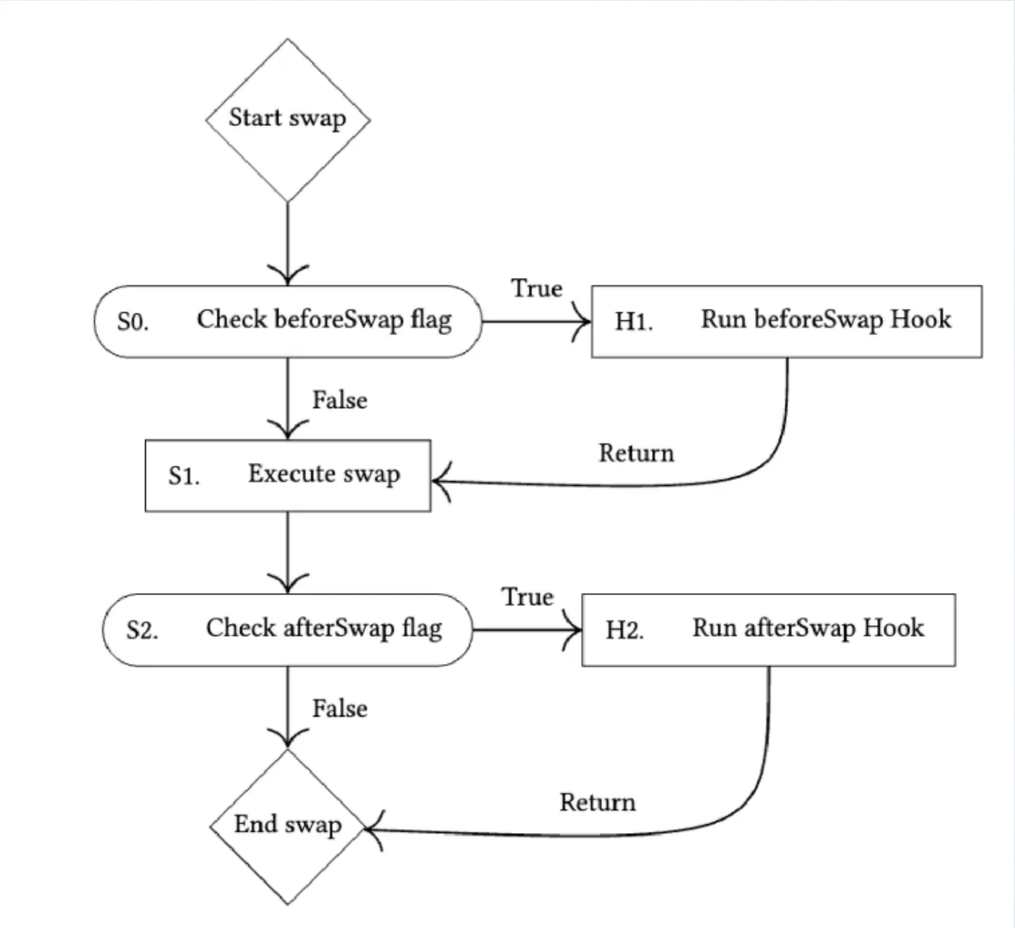
- （单例）合并所有的流动池，在一个合约里，不用部署新池，多跳兑换费用低。



UNISWAP V4

- Hook - 允许池创建者添加额外逻辑，特定新功能， 如：
  - 动态手续费、KYC 验证、自动复利 LP 手续费





UNISWAP X

- 背景：（DeFi）市场中流动性分散和交易路由复杂，用户难以获得最佳价格。
- UniswapX 旨在通过聚合链上和链下流动性，简化路由并优化价格发现。
- UniswapX 引入了“意图驱动”的交易模式，允许用户通过签名订单表达其交易意图，基于荷兰拍卖（Dutch Auction）由竞争者（称为 Fillers）使用任意流动性来源执行这些订单，从而实现实现跨 AMM（自动做市商）和其他流动性来源的交易路由。
- UniswapX 支持无 gas 交易、MEV 保护和跨链兑换，提升用户体验。

其他的 DEX 协议

SushiSwap

PancakeSwap

Curve

0x

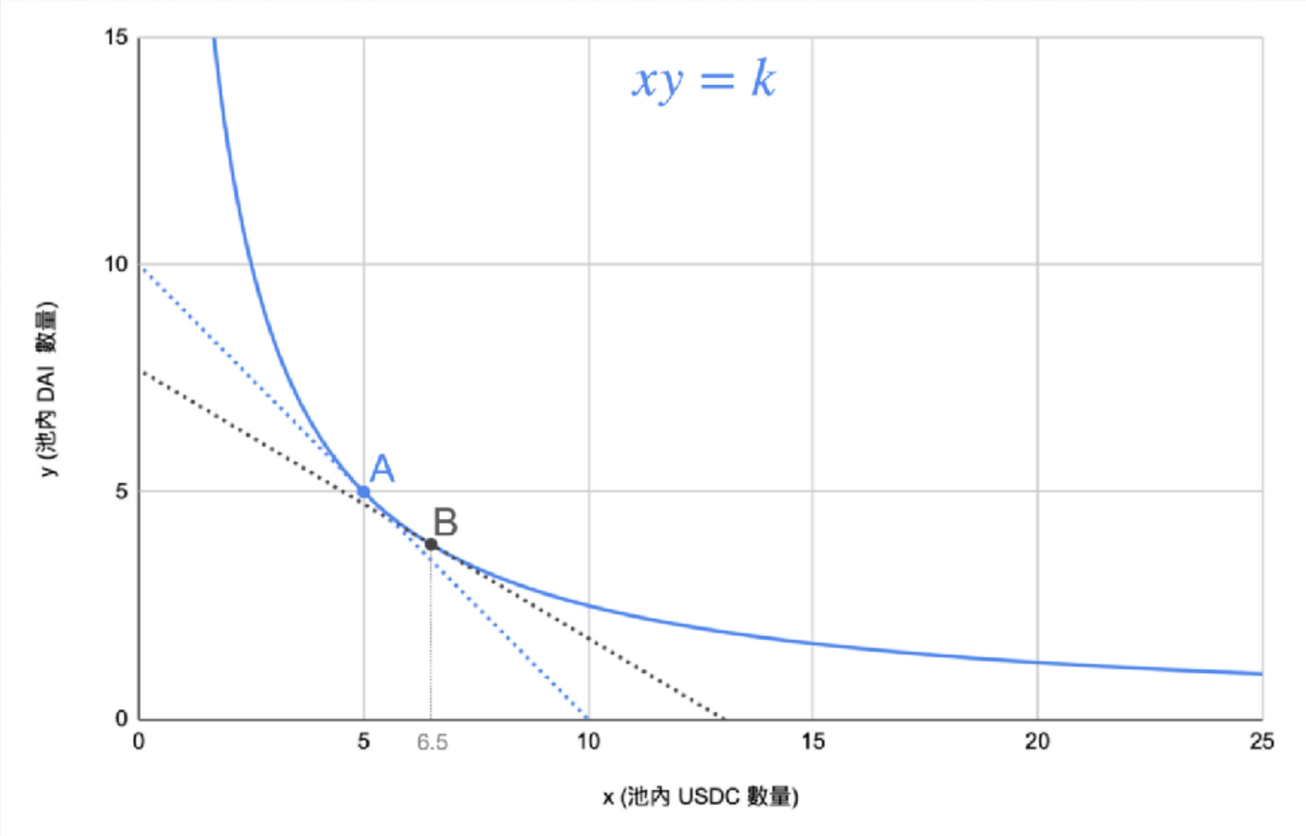
SushiSwap / PancakeSwap

- SushiSwap 早期很成功的 Uniswap 仿盘，对通过“吸血攻击”（vampire attack）吸引 Uniswap 用户
- Uniswap:
  - 提供流动性时才赚取资金池的交易费，撤回流动性不再获得相应的收入
- SushiSwap = Uniswap + 流动性挖矿
  - 流动性挖矿：
  - 1. 为流动性提供者奖励
  - 2. 锁定流动性、弥补流动性损失
  - 公平发行协议币（sushi）： 去中心化治理、sushi 代币可持续给参与者带来协议收益
- PancakeSwap : 在 BNB Chain 的 Uniswap 仿盘， 也借鉴了 SushiSwap 挖矿 , BNB 链上最大的 DEX

Curve 协议 – 稳定币兑换

- 稳定币兑换可以使用 xy=k?

要价格波动小（2%以内），
一次交易的数量需要小于池
内的 1%



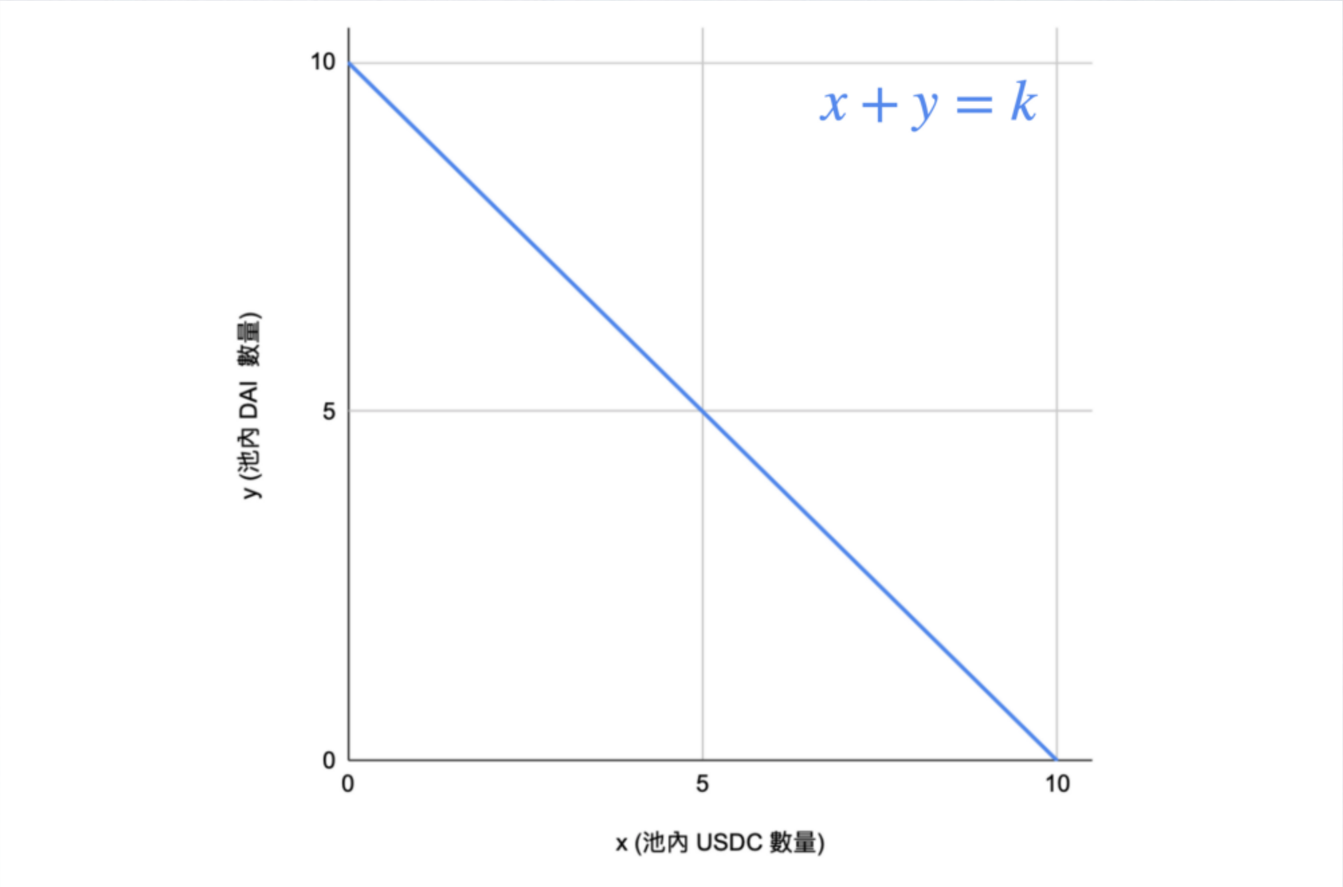
参考文章：https://medium.com/@cic.ethan/淺談穩定幣互換機制-從-balancer-到-curve-f638f29b33f9

Curve 协议

- 常量和模型 x + y=k

  优点：永远1:1兑换，没有滑点：
  ΔX = ΔY

  缺点：价格不应供需改变，出现以外部价差时，出现流动性枯竭。



Curve 协议

- 稳定币兑换

恒定乘积：双曲线，两端将接近无穷

恒定总和: 直线，0 滑点、流动性枯竭

$$x\,y = (\frac{D}{2})^2 \times \beta$$

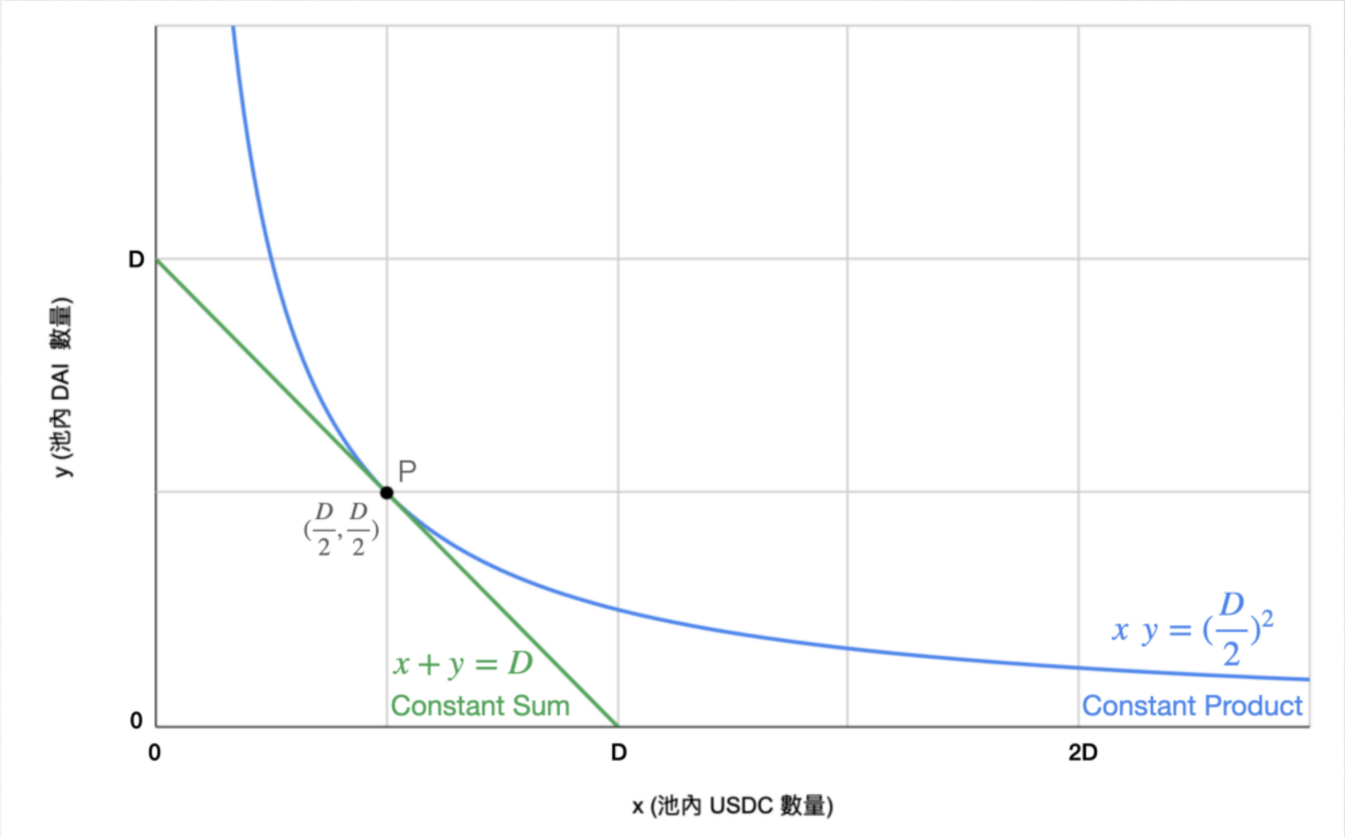
[数量]²

$$x + y = D \times \alpha D$$

[数量] [数量]

理想模型于两者之间，简单方式加权平均：

$$\alpha D(x+y) + \beta(xy) = \alpha D D + \beta(\frac{D}{2})^2$$

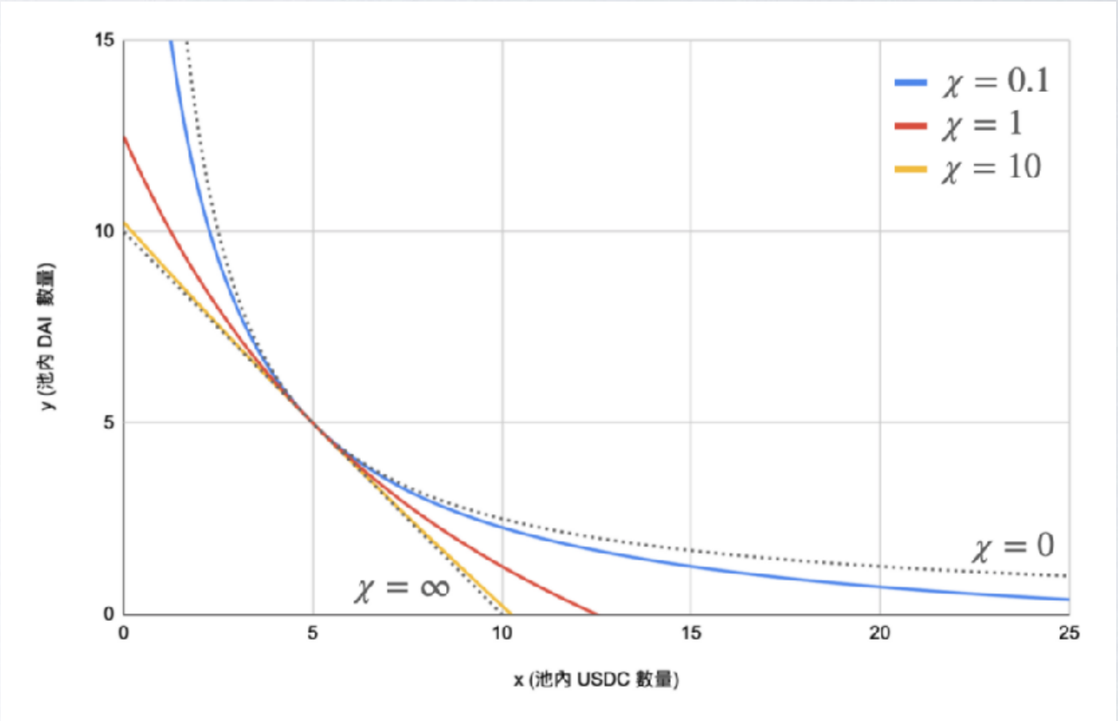


Curve 协议

- 稳定币兑换

$$\alpha D(x+y)+\beta(xy)=\alpha DD+\beta(\frac{D}{2})^2$$

$$\chi D(x+y)+(xy)=\chi D^2+(\frac{D}{2})^2 \quad ,\chi=\frac{\alpha}{\beta}$$



$\chi = 0$，方程退化为常量乘积模型， 当$\chi = \infty$，方程退化为常量和模型

可以通过设置 χ 来设计一条曲线

Curve 协议

- 引入动态参数

模型仍有一个缺陷：若希望价格越稳定，则曲线越接近直线，此時会面临流动性枯竭问题。能否在靠近平衡点时，χ 越大，远离平衡点时越小。

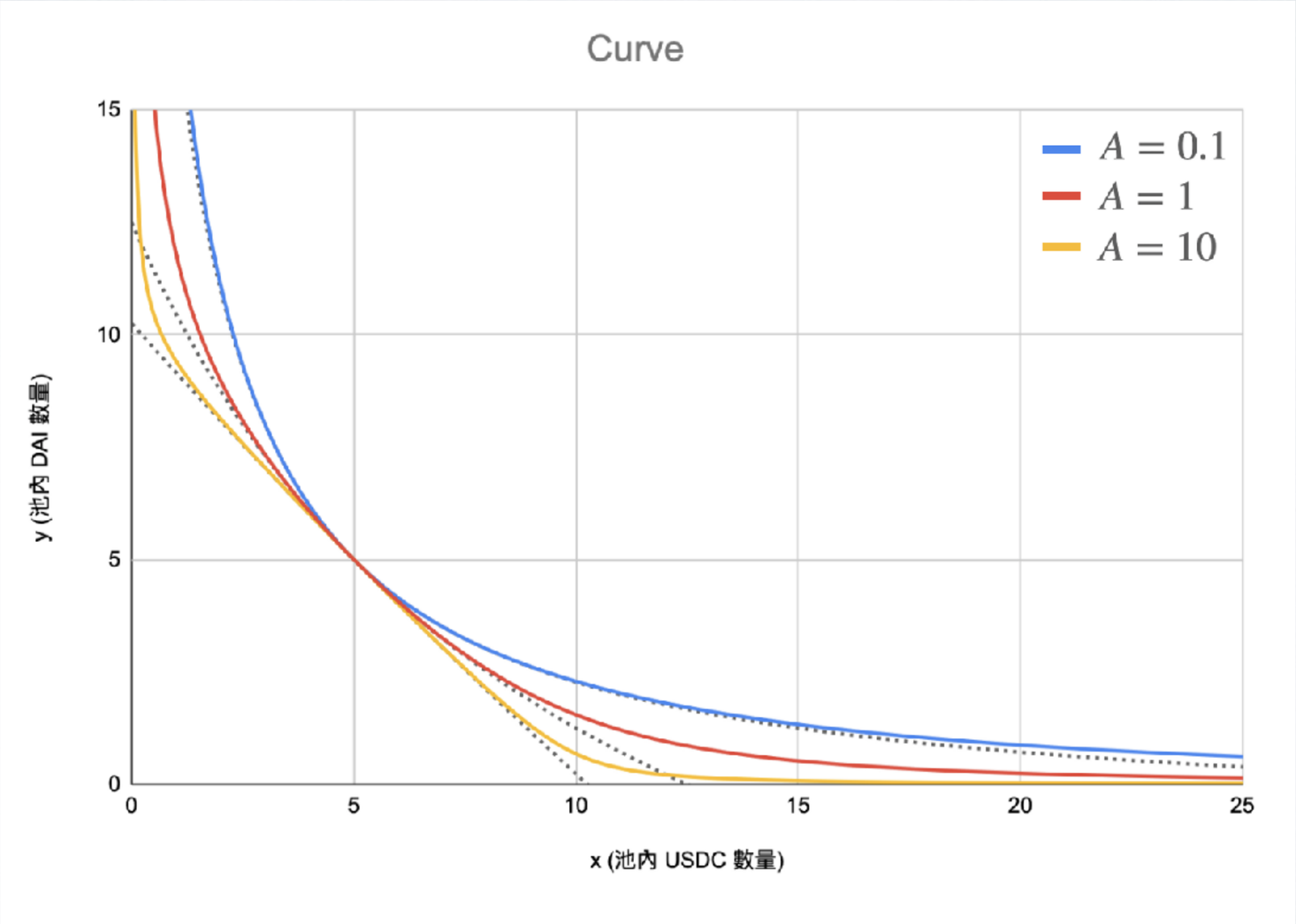
$$\chi = A\frac{xy}{(\frac{D}{2})^2}$$

A 为放大系数，决定平衡点附件的曲率，A 越大滑点越小。**蓝色部分为动态参数，在（0,1）之间：**
离平衡点越接近時，接近1，接近常量和模型。 离平衡点越远時，接近0，接近常量乘积模型。

$$A\frac{xy}{(\frac{D}{2})^2}D(x+y)+(xy)=A\frac{xy}{(\frac{D}{2})^2}D^2+(\frac{D}{2})^2$$

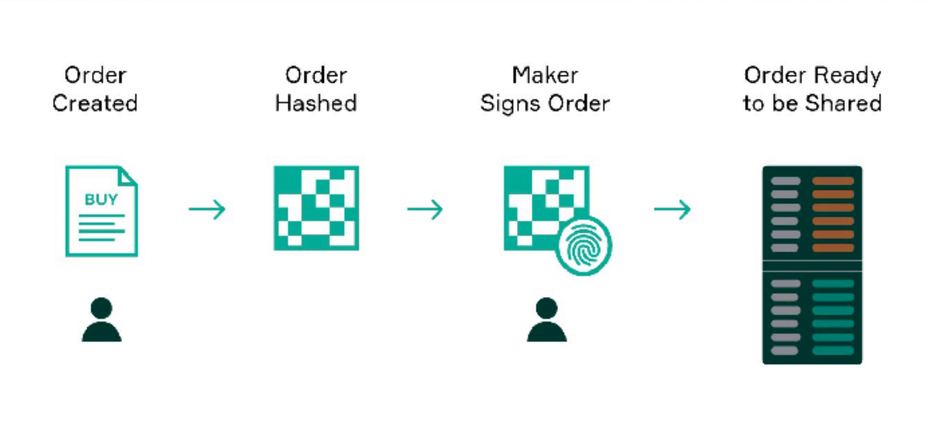
$$A2^2\ (x+y)+D=AD2^2+\frac{D^3}{2^2\ xy}$$

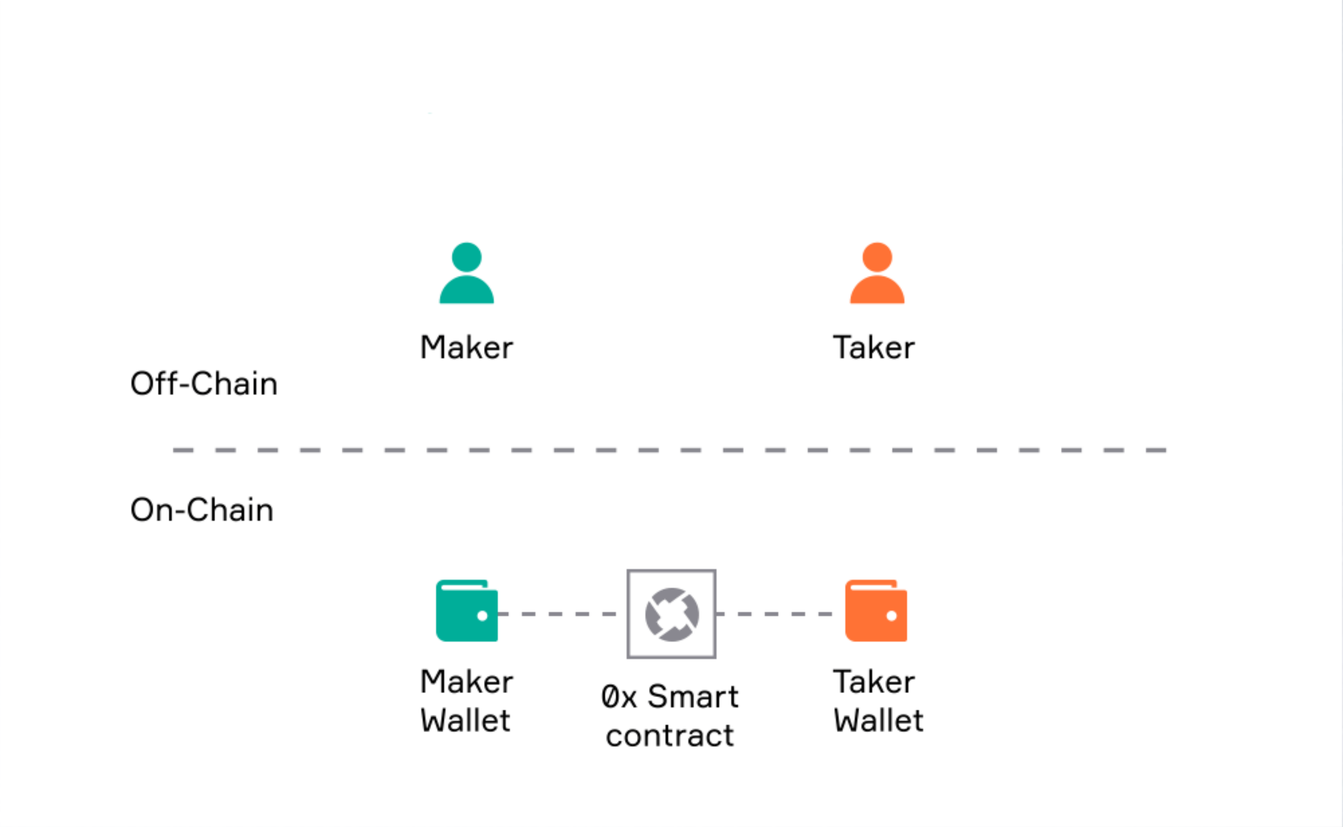
Curve 协议



0x 协议

- 链下撮合，链上结算





https://0x.org/docs/introduction/introduction-to-0x

交易聚合器

1inch　　Uniswap X　　OKX

1INCH

- 从多个 DEX（如 Uniswap、Sushiswap、Curve 等）中寻找最佳的交易路径和价格，为用户提供最优的兑换体验。
- 工作原理：
  - 把用户的 swap 请求拆分成多个子路径，分别在不同的 DEX 上执行。
  - 使用动态规划或优化算法（Pathfinder）寻找最佳分配方式。
  - 例如：你想兑换 1000 USDC -> ETH，1inch 会可能让你在：
    - 40% 在 Uniswap v3；
    - 30% 在 Sushiswap；
    - 30% 在 Curve；

      从而获得最好的最终结果。

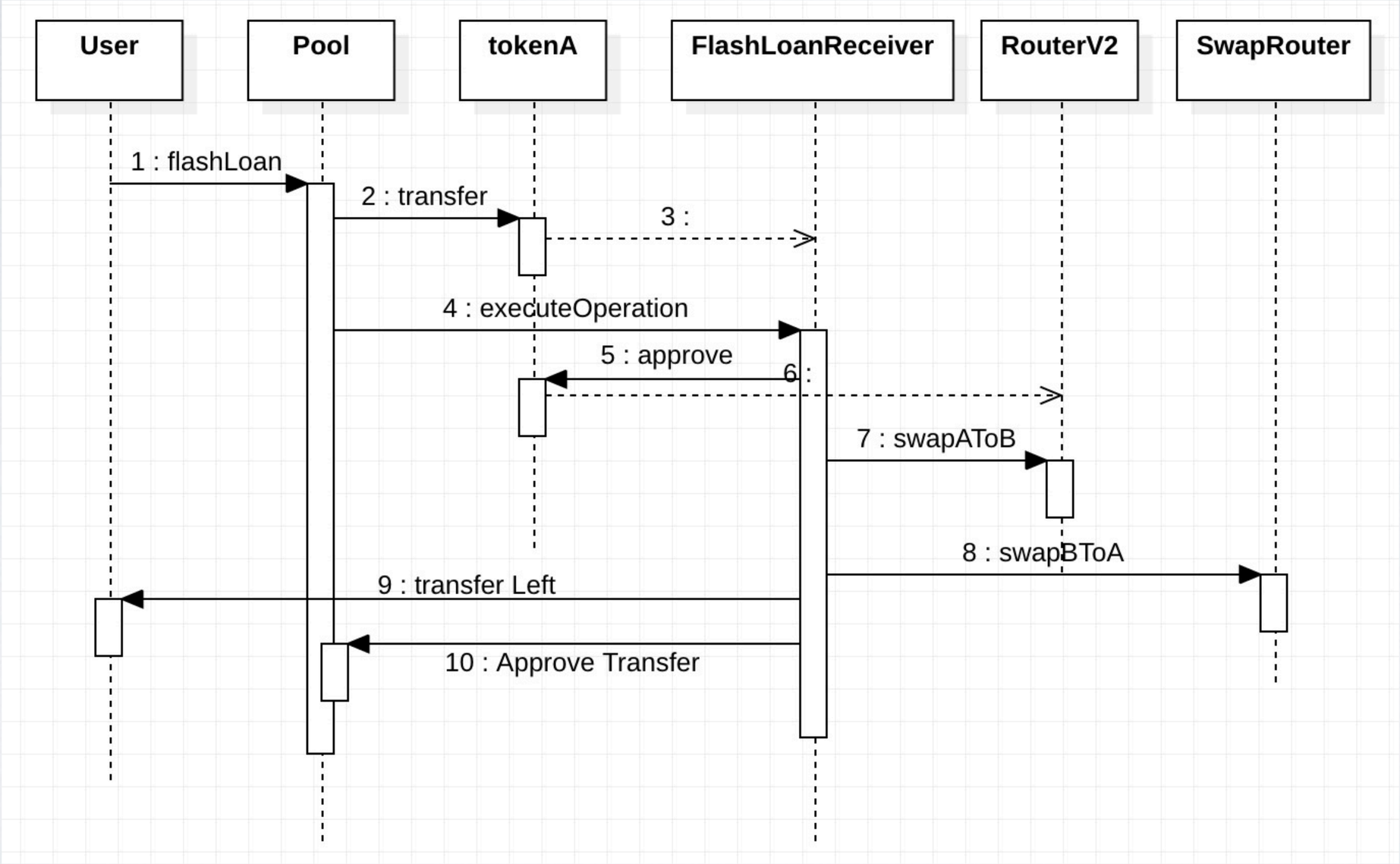
练习题

- 以太坊测试网上部署两个自己的 ERC20 合约MyToken，部署两个Uniswap，创建两个Uniswap V2 流动池（PoolA 和 PoolB），形成价差。
- 作业：编写合约执行闪电贷（参考 V2 的ExampleFlashSwap）：
  - uniswapV2Call中，用从 PoolA 收到的 TokenA 在PoolB 兑换为 TokenB 并还回到 uniswapV2 Pair 中。

https://decert.me/quests/2a63cf95-43ec-42ee-975f-2b41510492cd

练习题(可选)

- 在一笔交易中完成（模拟闪电贷交易）：
  - 在 AAVE 中借款token A
  - 使用token A在 Uniswap DEX 中交易兑换 token B， 然后在 SushiSwap 把 token B 兑换为 tokenA
  - tokenA 还款 给AAVE


User
Pool
tokenA
FlashLoanReceiver
RouterV2
SwapRouter
1 : flashLoan
2 : transfer
3 :
4 : executeOperation
5 : approve
6 :
7 : swapAToB
8 : swapBToA
9 : transfer Left
10 : Approve Transfer