

借贷

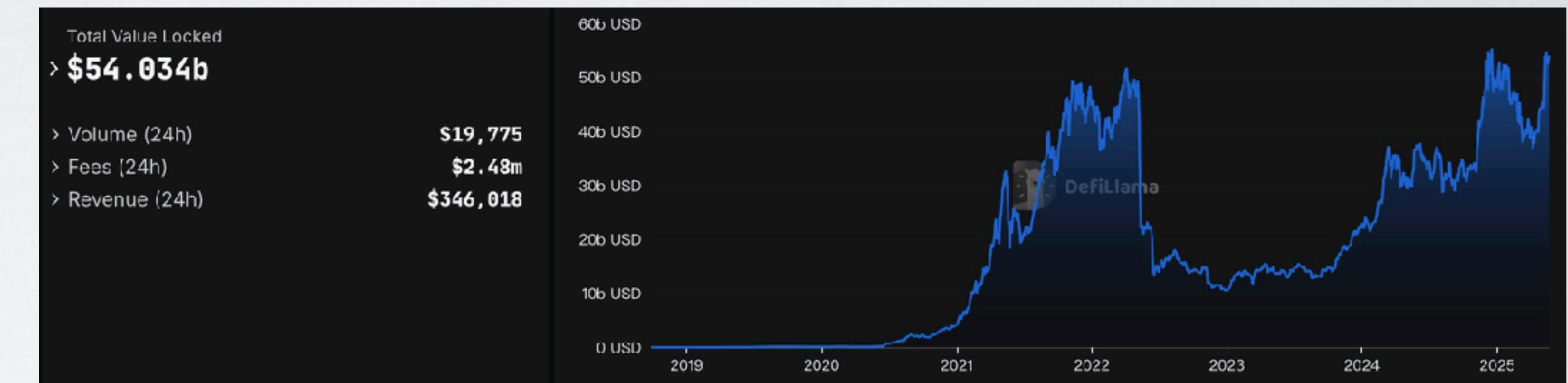
Tiny 熊

要点

- 借贷协议: 以 Compound 为例, 利率模型、清算
- Compound 协议演进及其他借贷协议

借贷协议

借贷是当前 DEFI 里
锁仓金额 (TVL) 最大的一个分类



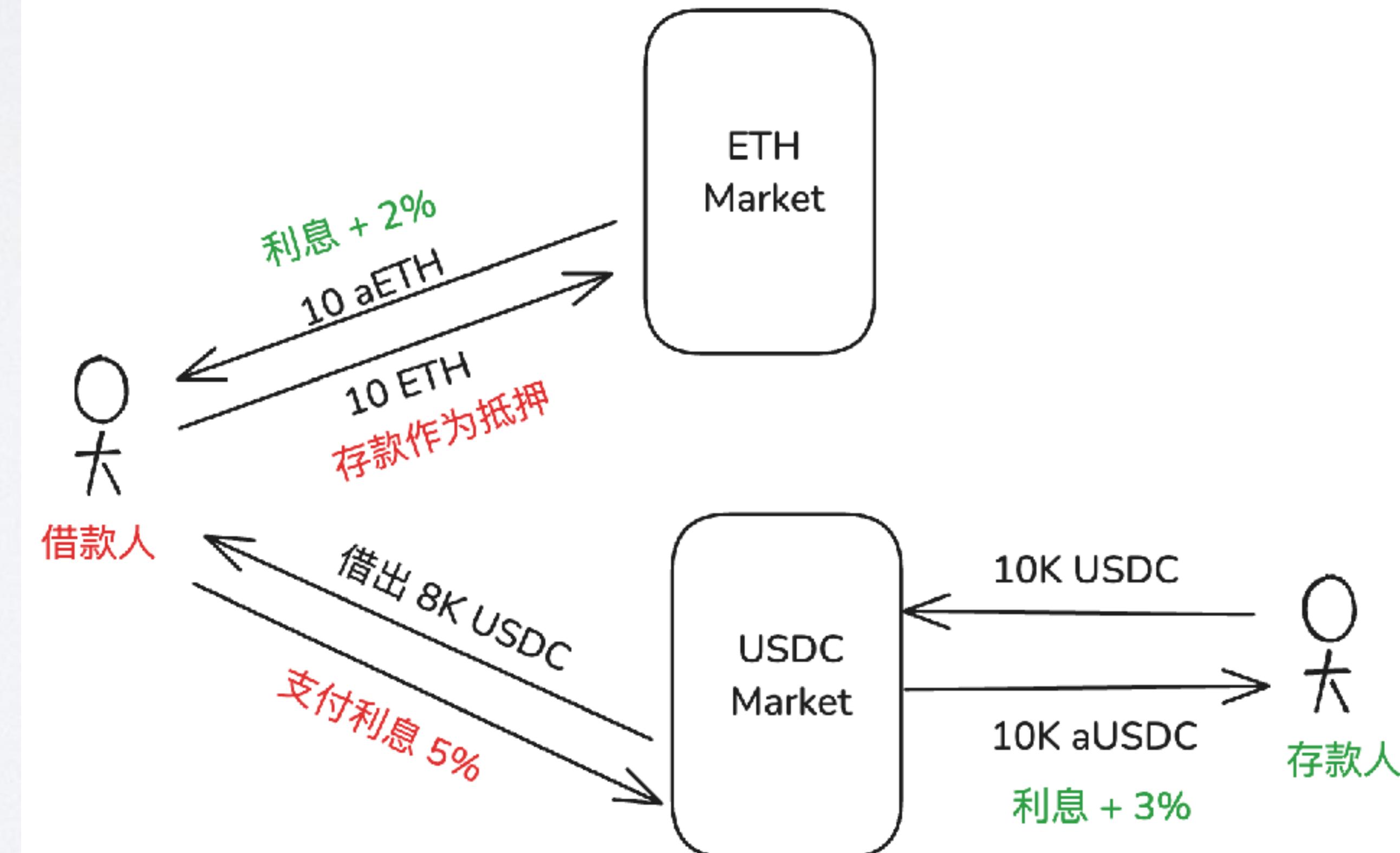
Protocol Rankings									
Rank	Compare	Name	1d Change	7d Change	1m Change	TVL	Fees 7d	Revenue 7d	
1	■	AAVE 16 chains	+0.15%	-2.63%	+32.51%	\$24.429b	\$9.89m	\$1.24m	
2	■	JustLend 1 chain	+0.66%	-0.64%	+43.83%	\$5.19b	\$107,510	\$6,671	
3	■	Morpho 18 chains	+0.56%	+5.70%	+33.77%	② \$3.495b	\$134,498		
4	■	SparkLend 2 chains	-2.09%	+6.63%	+48.60%	\$3.098b	\$1.1m	\$21,429	
5	■	Compound Finance 9 chains	-0.05%	-0.38%	+20.70%	\$2.638b	\$783,327	\$18,195	

借贷协议

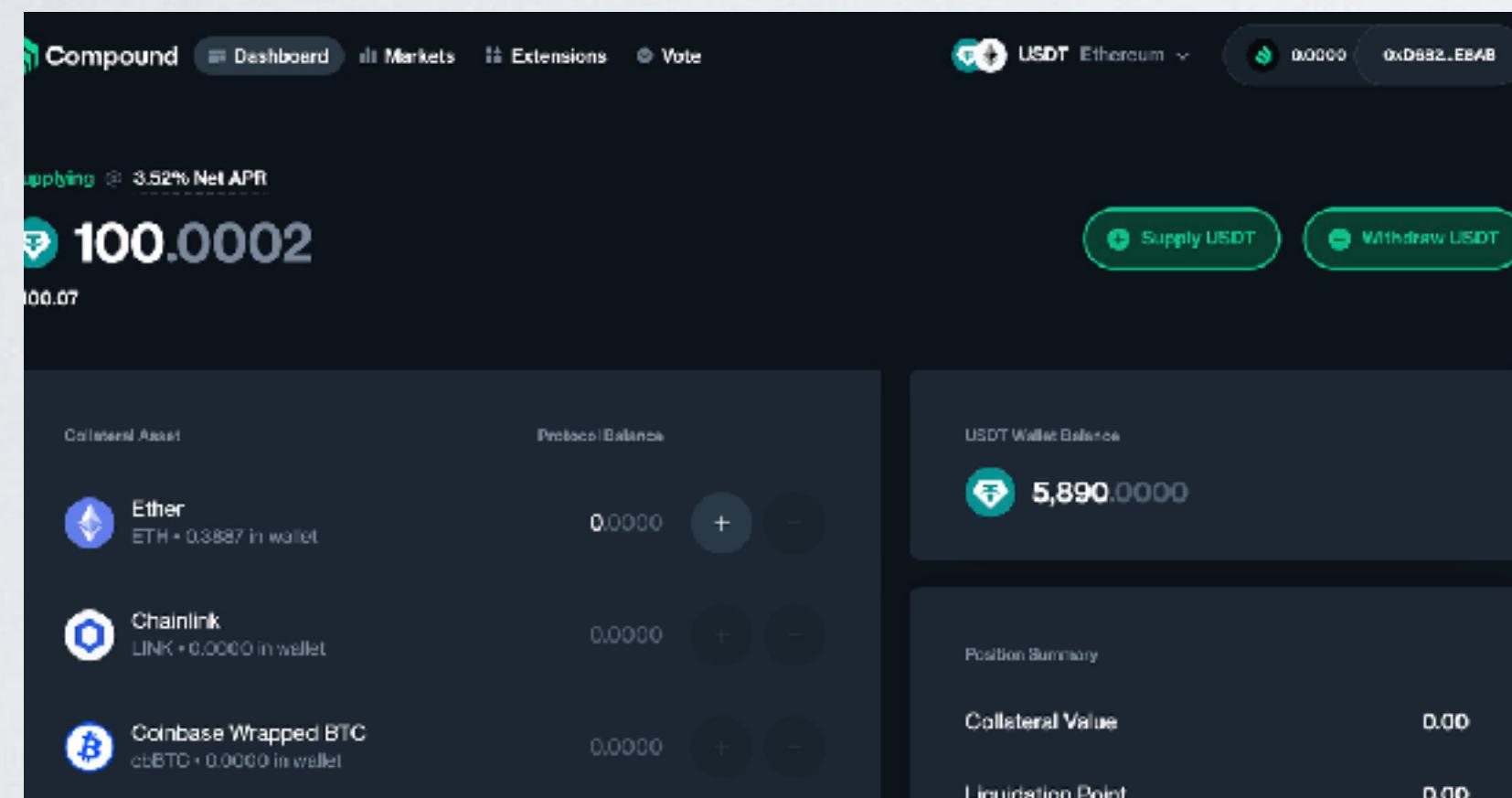
- 需要资金，但不想卖掉资产（看好未来升值潜力），就可以使用（超额）抵押借贷
- 主要以 Compound 和 Aave 为代表
 - Compound 最早（2018）推出池化机制和动态利率模型
 - Aave：引入闪电贷（Flash Loan） - 无抵押贷款（只要在同一笔交易中归还）

超额抵押

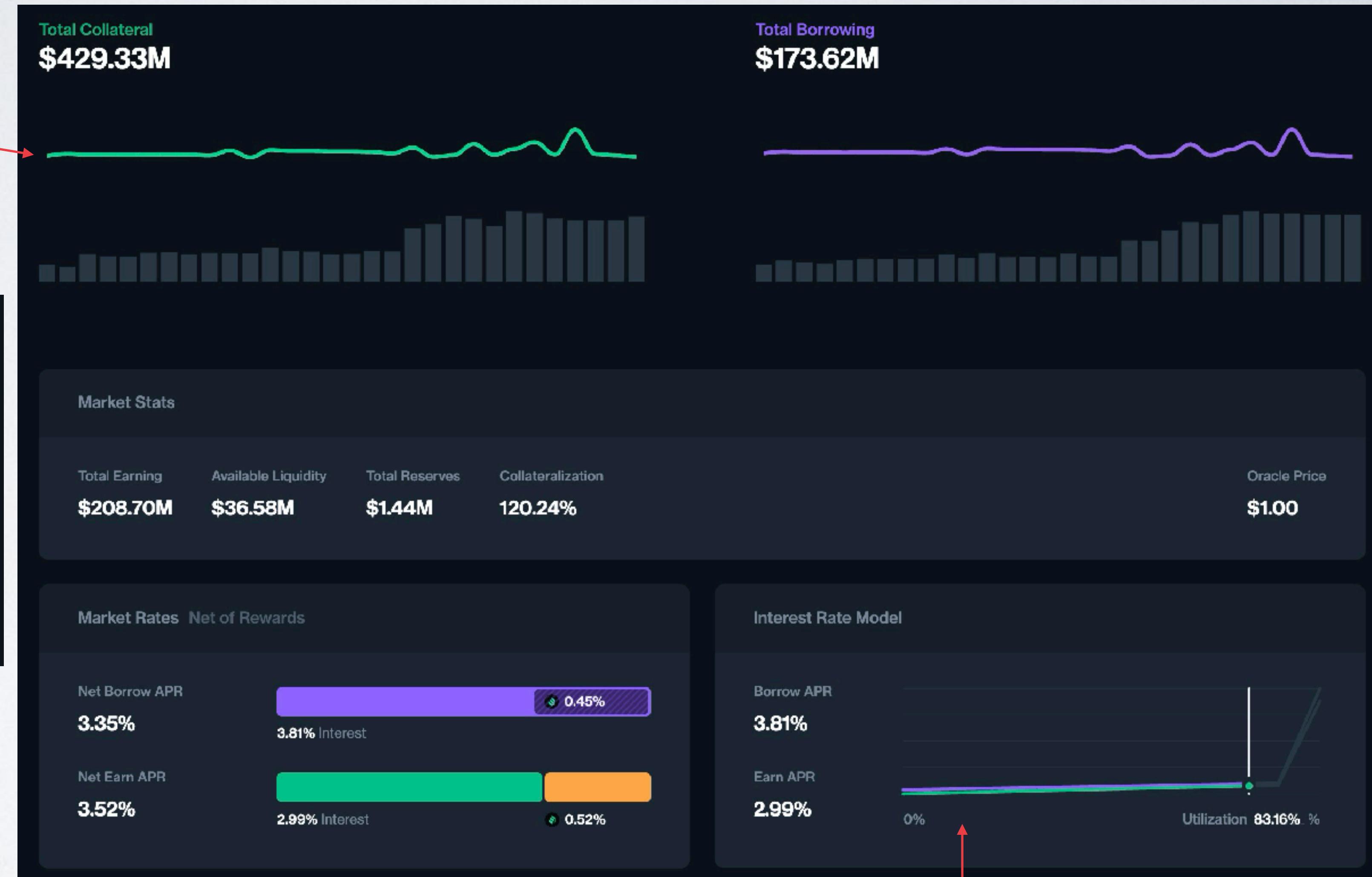
抵押品价值 >> 借出价值



演示



利率波动



利息计算方式(分为两个阶段)

<https://app.compound.finance/>

Compound - 利率模型

- 利率是实时动态的，根据资金的使用率 (Utilization Rate) 变化

$$U = \frac{\text{Total Borrows}}{\text{Total Cash} + \text{Total Borrows} - \text{Reserves}}$$

其中：

- Total Borrows**：当前池中所有借款总额
- Total Cash**：还未借出的资金
- Reserves**：协议预留的部分资金 (不可借出)

- 用户总共存入了: **1,000 USDC**
- 其中已经借出去: **800 USDC**
- 当前合约预留了: **20 USDC** (储备金)

$$U = \frac{800}{200 + 800 - 20} = \frac{800}{980} \approx 0.816 = 81.6\%$$

- 利率分为两个阶段，过临界点 (kink，如 0.8) 后利率迅速上升(吸引出借人，促使借款人还款)

$$\text{BorrowRate} = \begin{cases} \text{baseRate} + U \times \text{multiplier} & \text{if } U \leq \text{kink} \\ \text{baseRate} + \text{kink} \times \text{multiplier} + (U - \text{kink}) \times \text{jumpMultiplier} & \text{if } U > \text{kink} \end{cases}$$

$$\text{SupplyRate} = \text{BorrowRate} \times U \times (1 - \text{reserveFactor})$$

参数设置：
 baseRate : 2%
 Multiplier: 20%
 Kink: 80%
 JumpMultiplier: 150%
 reserveFactor: 10%

$$\begin{aligned} \text{BorrowRate} &= 0.02 + 0.80 \cdot 0.20 + (0.816 - 0.80) \cdot 1.50 \\ &= 0.02 + 0.16 + 0.024 = 0.204 = 20.4\% \text{ (年化借款利率)} \end{aligned}$$

$$\text{SupplyRate} = 0.204 \cdot 0.816 \cdot 0.90 \approx 0.1496 = 14.96\%$$

每个市场会有自己的参数设置

Compound - 利率模型

- 利率是实时动态的，在 Supply(获得 cToken)、Borrow、Redeem (赎回) 、Repay (还款) 动作是都会更新 (调用accrueInterest) 。
 - 每次交易的时候计算一次复利
- 如何计算每个人的结款利息和存款收益的呢？

$$\text{borrowIndex}_{\text{new}} = \text{borrowIndex}_{\text{old}} \times (1 + \text{borrowRate} \cdot \text{blockDelta})$$

$$\text{currentBorrow} = \text{principal} \times \frac{\text{borrowIndex}}{\text{accountBorrowIndex}}$$

```
// CToken.sol
    function accrueInterest() virtual override public returns (uint) {
// simpleInterestFactor = 每区块的借款利率 * blockDelta
// borrowIndexNew (利息累计因子) = simpleInterestFactor * borrowIndex + borrowIndex
// 累计的利息 = simpleInterestFactor * totalBorrows
// totalBorrowsNew = 累计的利息 + totalBorrows
// totalReservesNew (新储备) = 累计的利息 * reserveFactor + totalReserves
}
```

Compound - 利率模型

$$\text{borrowIndex}_{\text{new}} = \text{borrowIndex}_{\text{old}} \times (1 + \text{borrowRate} \cdot \text{blockDelta})$$

$$\text{currentBorrow} = \text{principal} \times \frac{\text{borrowIndex}}{\text{accountBorrowIndex}}$$

举例：

初始：

- 你借了 100 USDC
- `accountBorrowIndex = 1e18`
- 当前 `borrowIndex = 1e18`

$$\rightarrow \text{当前负债} = 100 \times \frac{1e18}{1e18} = 100 \text{ USDC}$$

1个月后：

- 系统 `borrowIndex` 增长到了 $1.02e18$
- 你还没还款

$$\rightarrow \text{当前负债} = 100 \times \frac{1.02e18}{1e18} = 102 \text{ USDC}$$

(即 2 USDC 的利息)

Compound - cToken

- 存款的利息是通过 cToken 来体现的。存入 Token 获得 cToken (类似 Vault)
- cToken: 是存入资产的凭证, 也是生息代币, 每一种标的资产都有对应的一种 cToken, 比如, ETH 对应 cETH, USDT 对应 cUSDT, 生息体现为 cToken 与 Token 的兑换比例不断变大。
- 标的资产 (Underlying Token) : 借贷资产, 每种标的资产都有一个抵押因子 (Collateral Factor), 代表用户抵押的资产价值对应可得到的借款的比率

$$\text{exchangeRate} = \frac{\text{Total Cash} + \text{Total Borrows} - \text{Total Reserves}}{\text{Total Supply of cToken}}$$

每次`accrueInterest()` 分子都会增加

举例:

初始状态:

- 你存入 100 USDC → 获得 5000 cUSDC
- `exchangeRate = 0.02` (1 cUSDC = 0.02 USDC)

1 个月后 (利息增长):

- 借款人支付利息, `exchangeRate` 涨到 0.021

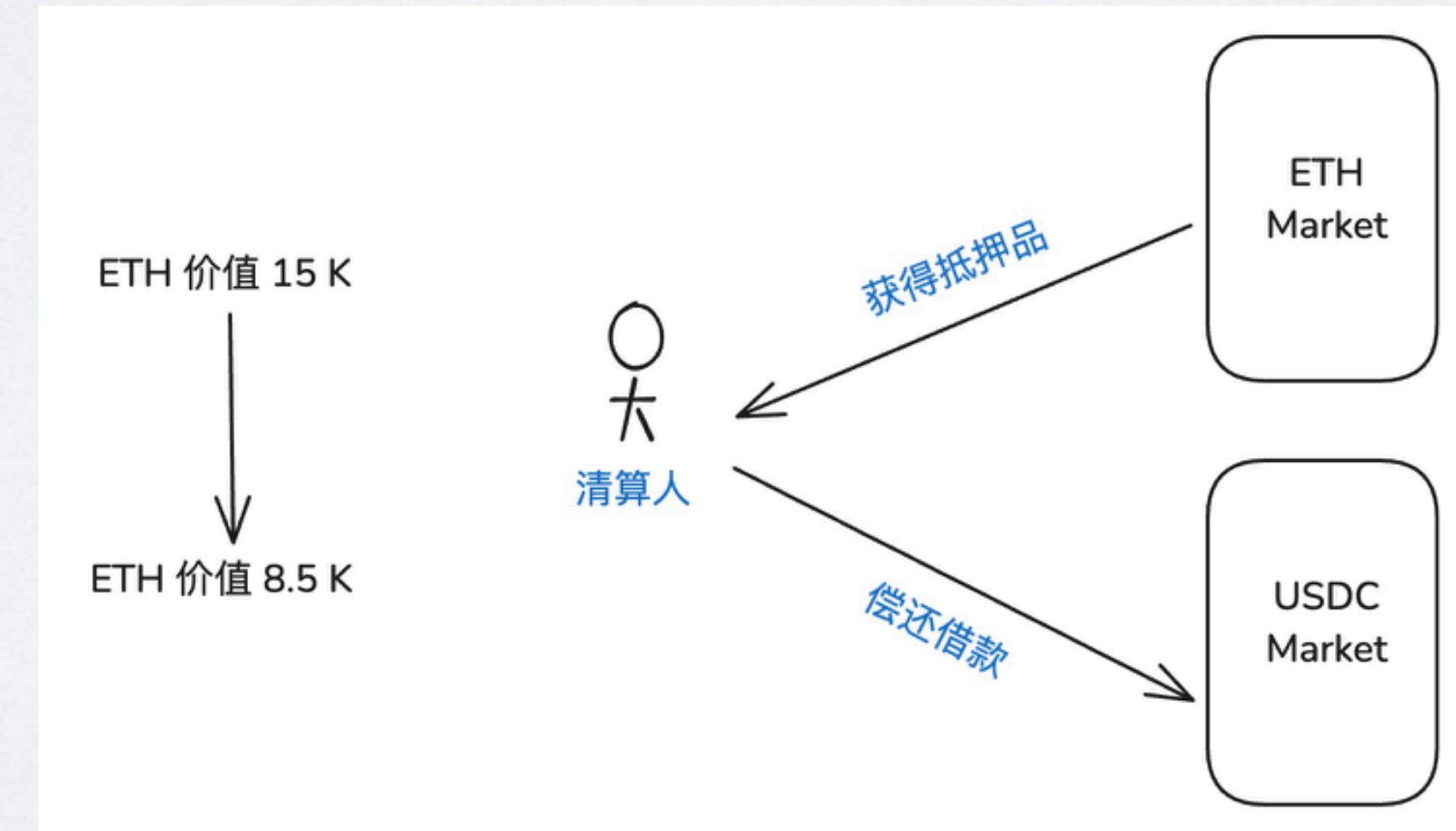
你没有任何操作, 但你的资产变为:

$$5000 \times 0.021 = 105USDC$$

借贷协议

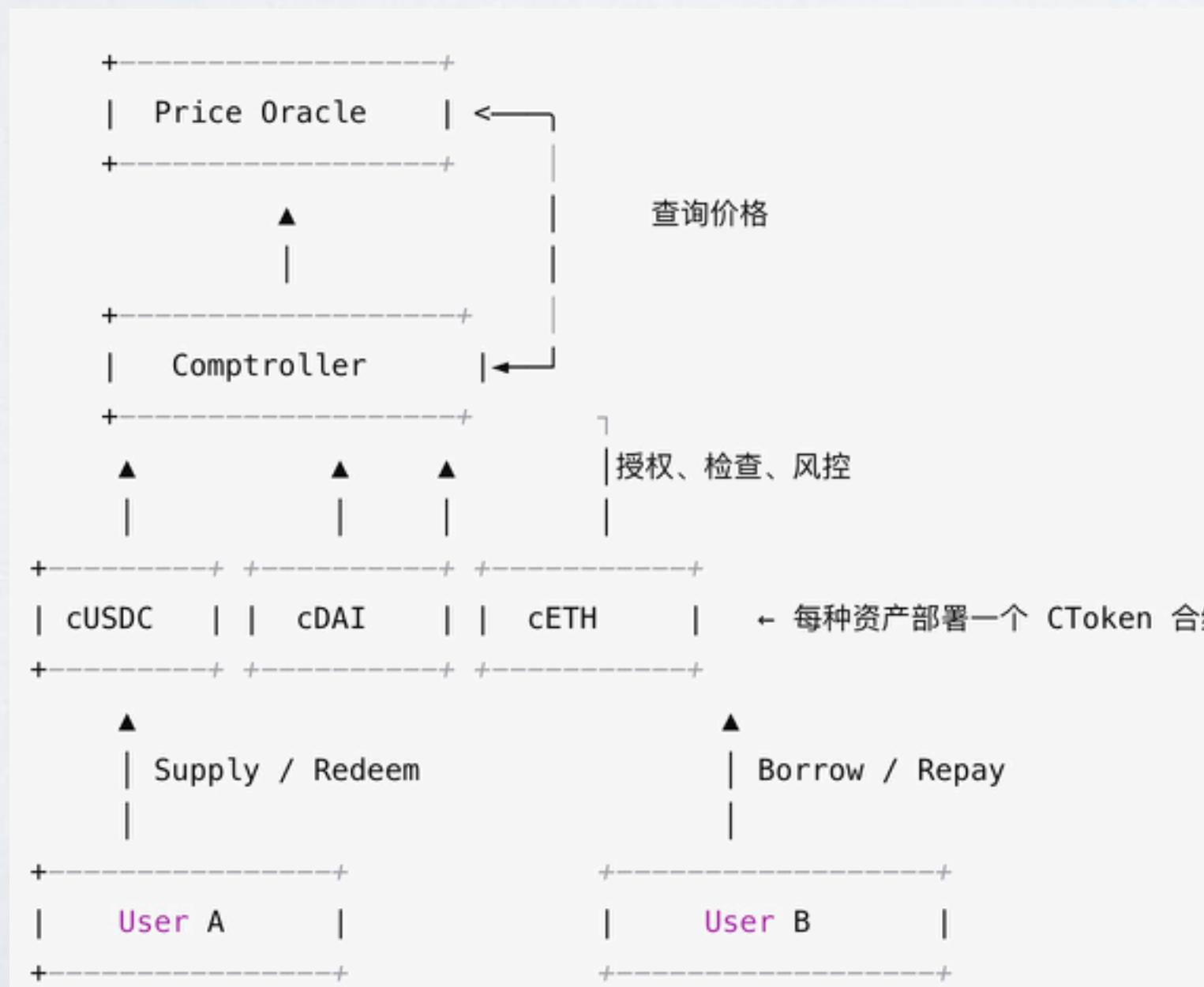
- 借贷协议通常还有两个重要的组成部分：

- 预言机(Oracle)：把现实世界的信息（资产价格）传递给链上，借贷协议利用预言机获取资产价格，以便计算抵押品和借款价值
- 清算：当抵押品价值接近借出资产价值时，第三方可替借款偿还借款并获得清算奖励（折扣）。



Compound (v2) - 预言机方案

- 获取 Chainlink price feeds 或其他数据源
- Compound Labs 运营的“Poster”账户将价格定期签名并发布到合约 (open-oracle)
- 获取价值时使用 `oracle.getUnderlyingPrice(cToken)`: 使用 **ETH 币本位**
- Compound 的 Comptroller 是一个统一的风险控制器: 在用户借款、赎回、清算时统计资产价值总额



借款流程:

1. ➤ `cUSDC`:
 - 调用 `accrueInterest()`, 更新利息
 - 调用 `Comptroller.borrowAllowed(...)` 进行权限检查
2. ➤ `Comptroller`:
 - 查询账户流动性和价格 (调用 Oracle)
 - 判断是否超出抵押限额
3. ➤ `cUSDC`:
 - 增加借款记录
 - 将 USDC 发送到用户钱包

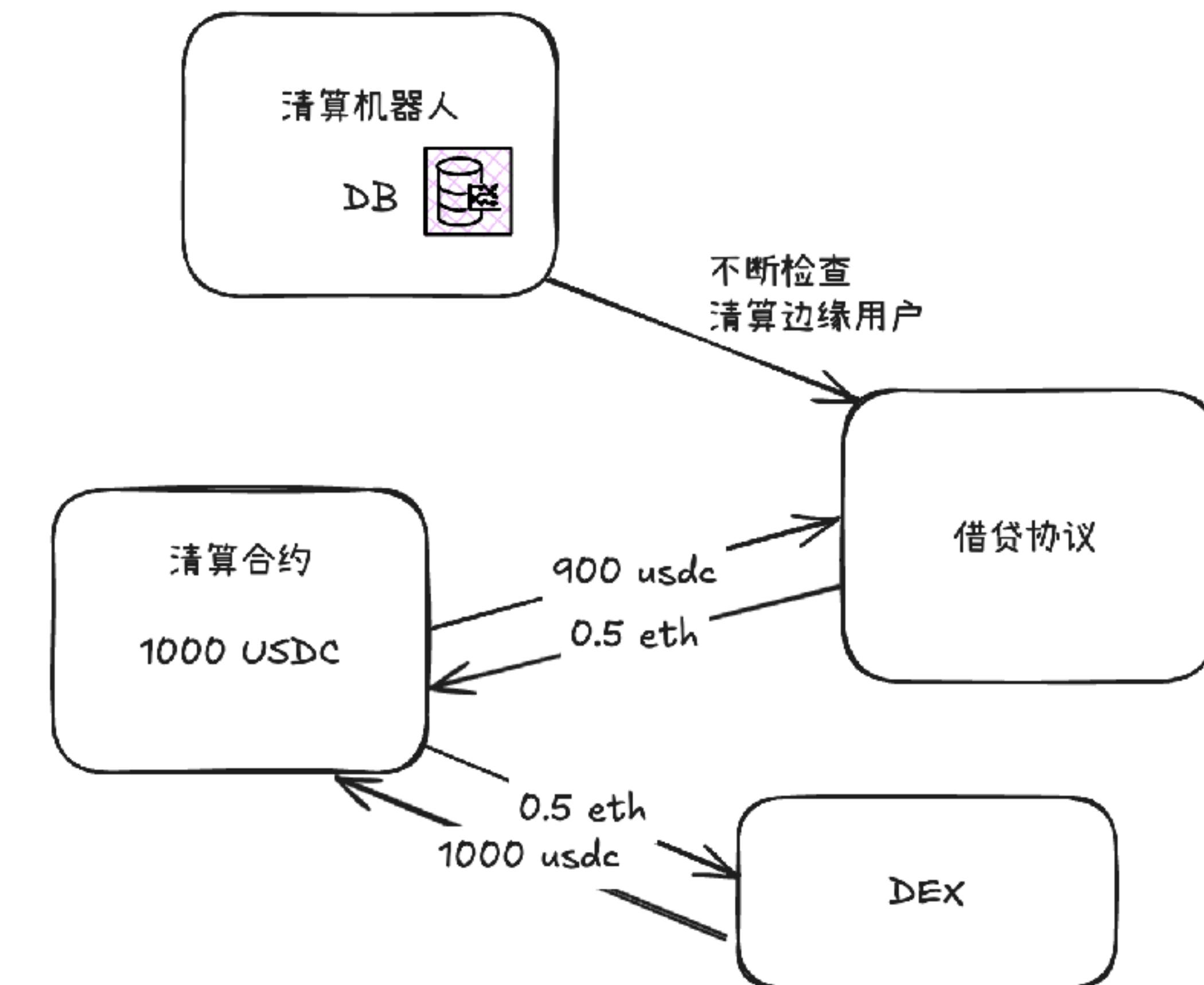
清算 (Liquidation)

- 及时清算可保护协议不会因坏账亏损，确保协议安全，清算可获得奖励，激励自发维护协议安全。
- 当 Comptroller.getAccountLiquidity() 根据抵押率 (collateral factor) 计算用户的账户健康度 - liquidity: 净可借额度, shortfall: 不足额度 (> 0 可以被清算)
- 抵押率 (collateral factor) : 抵押的资产用来借款的比例, 例如: 抵押率为 75%, 抵押了价值 \$1000, 最多可借 \$750, 当借款价值 > \$750, 变得不健康, 可以被清算。 (当有多个抵押资产时, 由清算人选择其中一个清算)



清算机器人

清算收益是无风险收益
被 MEV 机器人抢



Compound v2 合约关系

- CToken: cToken合约是主要业务逻辑的合约, CToken合约引入了 InterestRateModel 合约和 Comptroller 合约
- InterestRateModel: 利率模型的抽象合约, JumpRateModelV2、JumpRateModel、WhitePaperInterestRateModel 都是具体的利率模型实现。
- Comptroller: 安全检验, 获取 PriceOracle 的价格
- PriceOracle: 预言机抽象合约, SimplePriceOracle 实现
- Governance: 治理合约, Token: Comp

<https://github.com/compound-finance/compound-protocol>

Compound 协议 - 演进

- Compound V2：
 - 池化风险，如果某种资产被恶意操控价格，可能导致整个池子的资金耗尽。
 - 如果用户存入的资产被借走，用户可能需要等待其他用户归还才能提取。
 - 清算链式反应：若某资产价格剧烈波动，清算可能会引发连锁反应
- Compound V3（代号“Comet”）：使用单一借贷资产模式（瘦身，以提升安全性）
 - 区分抵押资产和基础资产，抵押资产不可以借出，也不产生利息
 - 以 USDC 池为例，ETH、WBTC 等资产仅作为抵押资产，仅有 USDC 可以借出，且赚取利息
 - 每个基础资产池，有自己的抵押品参数设置(硬顶供应)

其他借贷协议

- AAVE v2 - 闪电贷， 在一个交易里，可以先借后还， 借的资金可以在其他协议套利
 - 调用 `lendingPool.flashLoan` 后，会回调用户实现的 `executeOperation()`， 用户可在 `executeOperation` 函数中套利
- AAVE v3 - 跨链借贷
- Morpho：允许用户创建独立的借贷市场，每个市场由单一抵押资产（如 ETH）和单一借款资产（如 USDC）组成。降低治理需求，创建市场后，利率模型、预言机等参数便不可修改。
 - 每个借贷市场独立运行，风险不会波及其他市场
 - 引入 p2p 模式，用户存入或借出资产，会尝试从现有的相反交易方向中“直接撮合”。