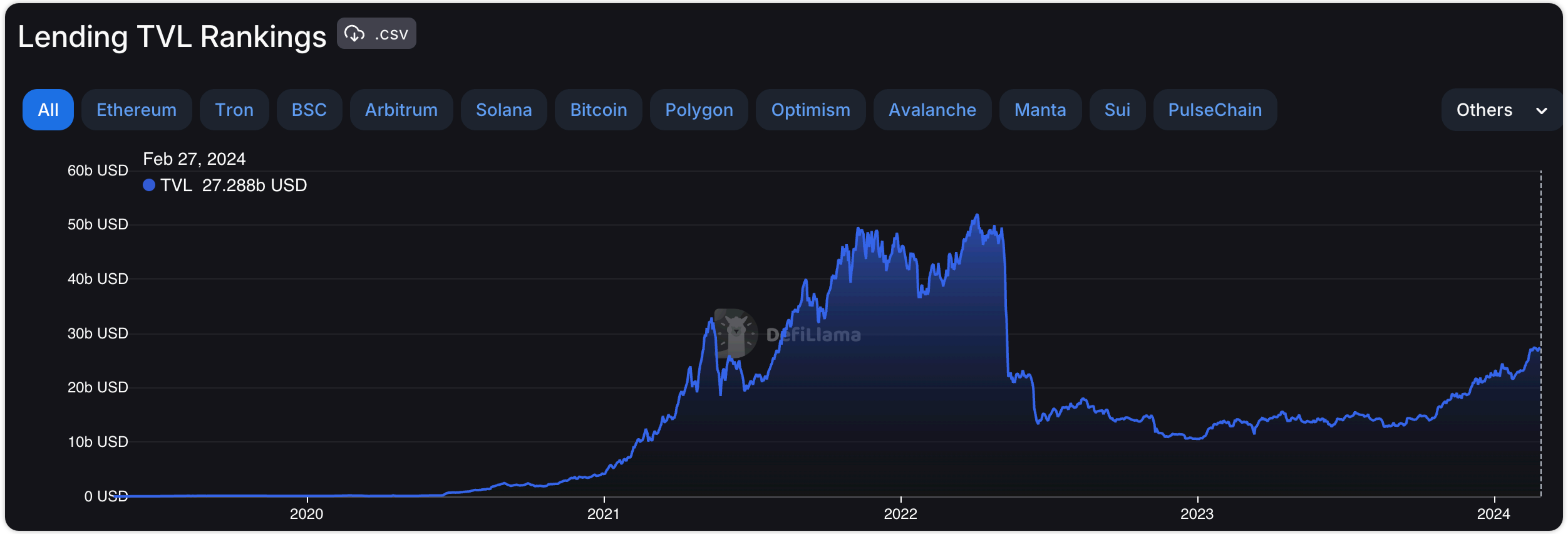
Compound

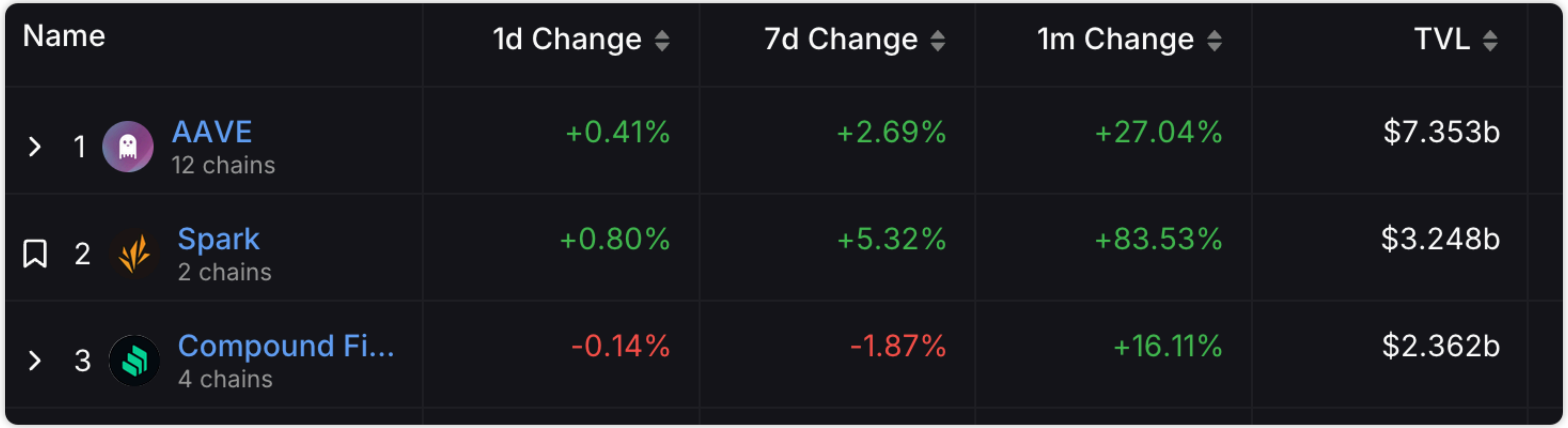
Decentralized Finance Study Case

by 0xqige.eth

Lending Protocol



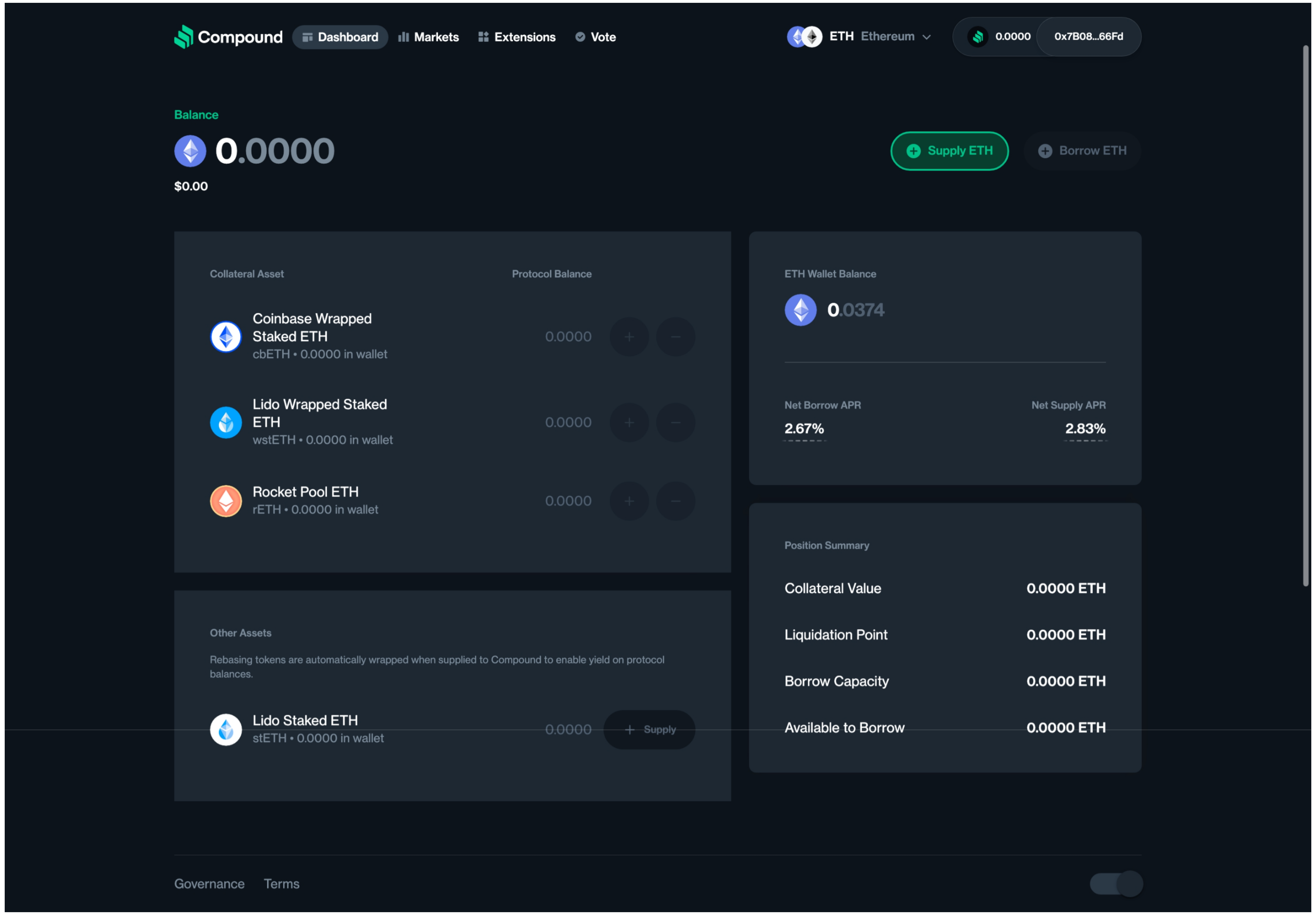
Lending TVL Rankings



Name	1d Change	7d Change	1m Change	TVL
1 AAVE 12 chains	+0.41%	+2.69%	+27.04%	$7.353b
2 Spark 2 chains	+0.80%	+5.32%	+83.53%	$3.248b
3 Compound Fi... 4 chains	-0.14%	-1.87%	+16.11%	$2.362b

What is Compound

a DeFi borrowing and lending protocol



Balance

0.0000

$0.00

Collateral Asset	Protocol Balance
Coinbase Wrapped Staked ETH cbETH • 0.0000 in wallet	0.0000
Lido Wrapped Staked ETH wstETH • 0.0000 in wallet	0.0000
Rocket Pool ETH rETH • 0.0000 in wallet	0.0000

Other Assets

Rebasing tokens are automatically wrapped when supplied to Compound to enable yield on protocol balances.

Lido Staked ETH
stETH • 0.0000 in wallet — 0.0000

ETH Wallet Balance

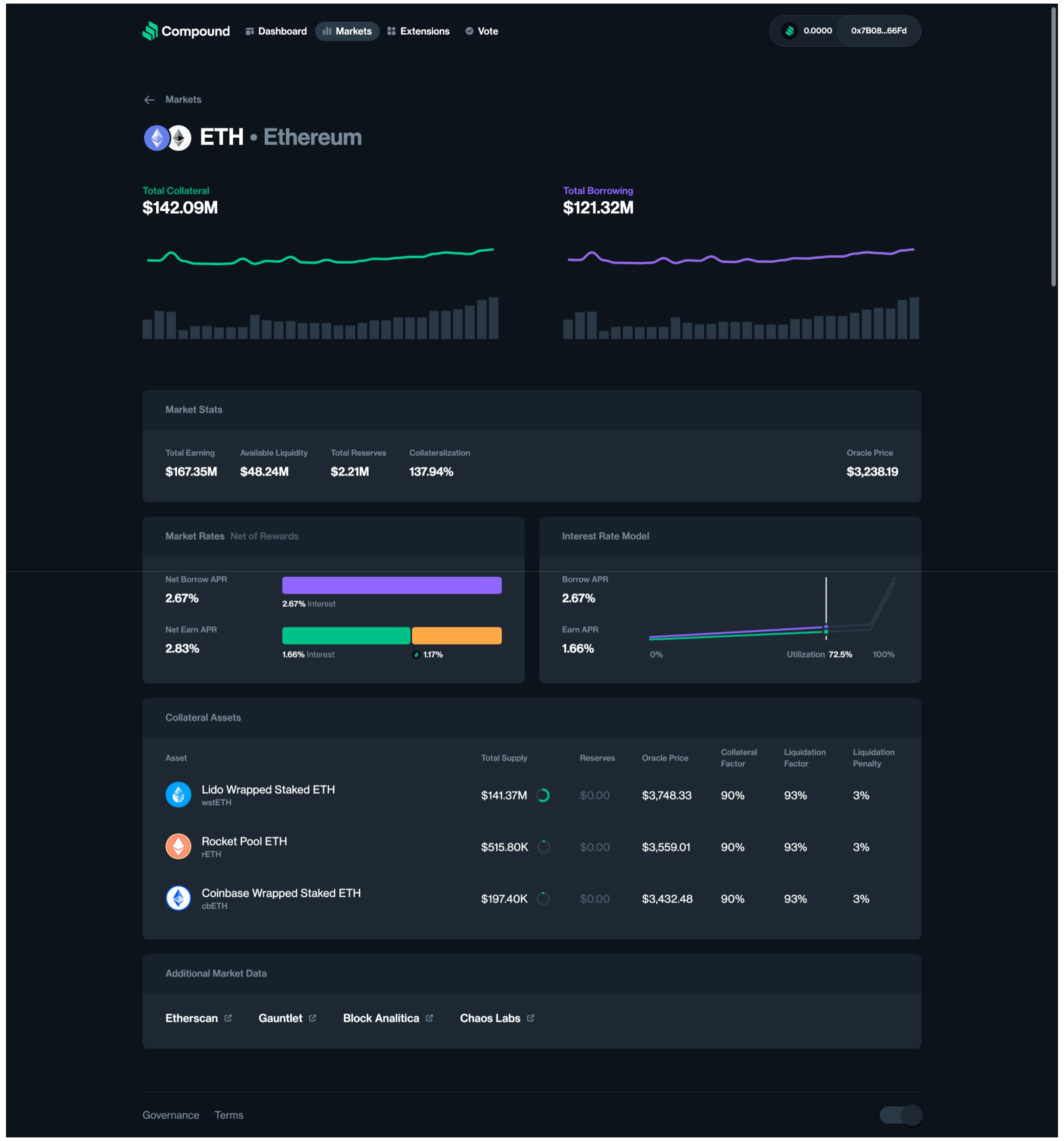
0.0374

Net Borrow APR 2.67%

Net Supply APR 2.83%

Position Summary

Collateral Value	0.0000 ETH
Liquidation Point	0.0000 ETH
Borrow Capacity	0.0000 ETH
Available to Borrow	0.0000 ETH



ETH • Ethereum

Total Collateral $142.09M

Total Borrowing $121.32M

Market Stats

Total Earning	Available Liquidity	Total Reserves	Collateralization	Oracle Price
$167.35M	$48.24M	$2.21M	137.94%	$3,238.19

Market Rates Net of Rewards

Net Borrow APR 2.67% — 2.67% Interest

Net Earn APR 2.83% — 1.66% Interest, 1.17%

Interest Rate Model

Borrow APR 2.67%

Earn APR 1.66%

Collateral Assets

Asset	Total Supply	Reserves	Oracle Price	Collateral Factor	Liquidation Factor	Liquidation Penalty
Lido Wrapped Staked ETH wstETH	$141.37M	$0.00	$3,748.33	90%	93%	3%
Rocket Pool ETH rETH	$515.80K	$0.00	$3,559.01	90%	93%	3%
Coinbase Wrapped Staked ETH cbETH	$197.40K	$0.00	$3,432.48	90%	93%	3%

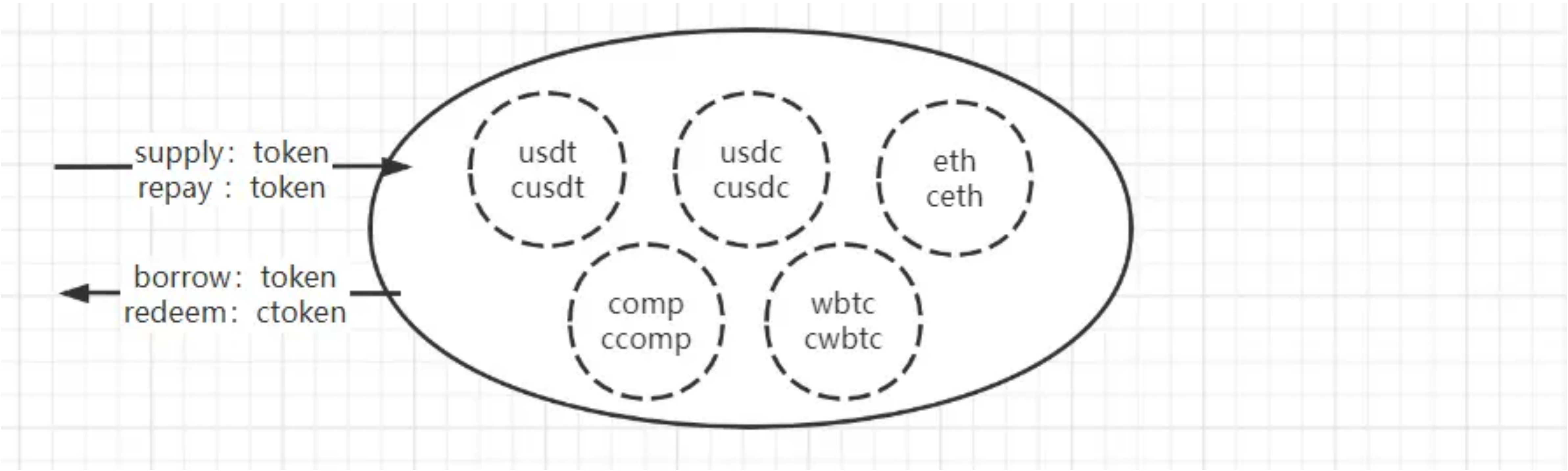
Additional Market Data

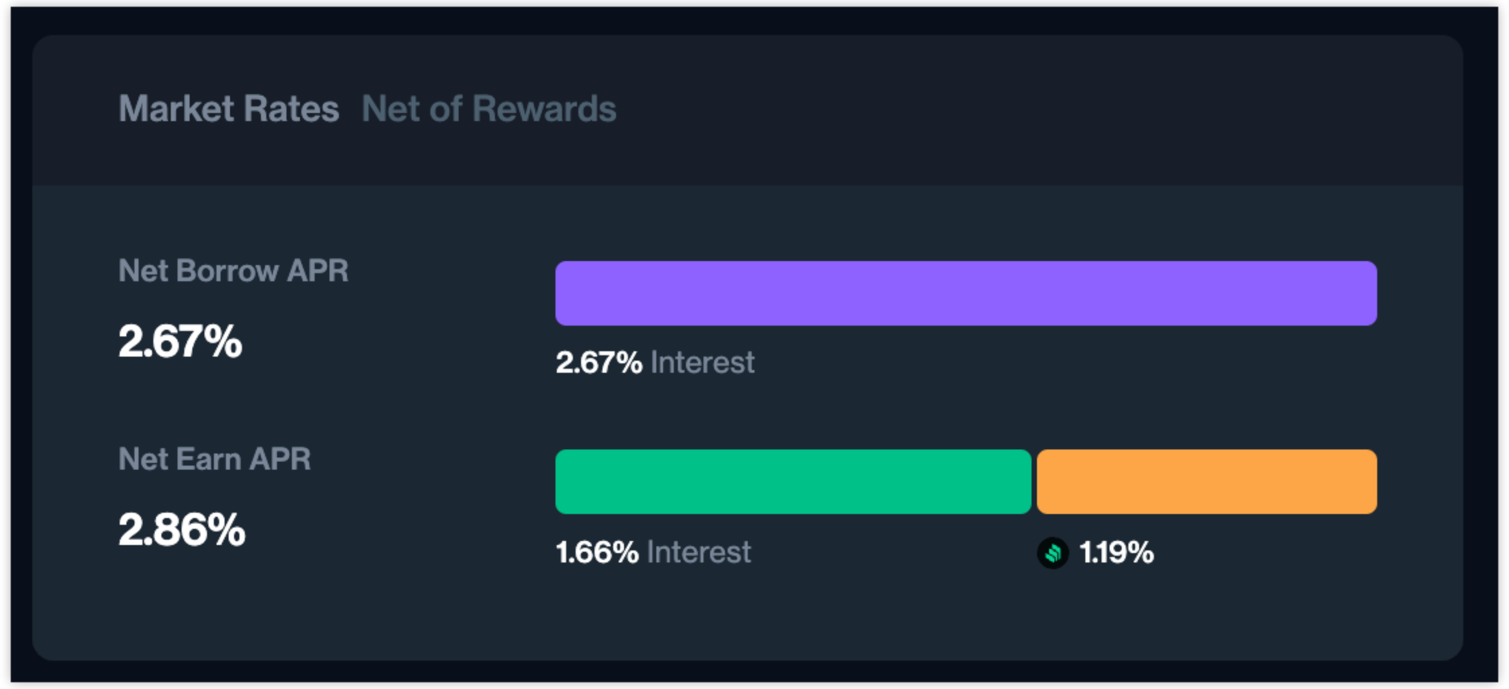
How work

Compound

Lenders: Deposit asset —> earning interest

Borrowers: Collateral asset -> borrow asset —> pay interest




Market Rates Net of Rewards

Net Borrow APR
2.67%
2.67% Interest

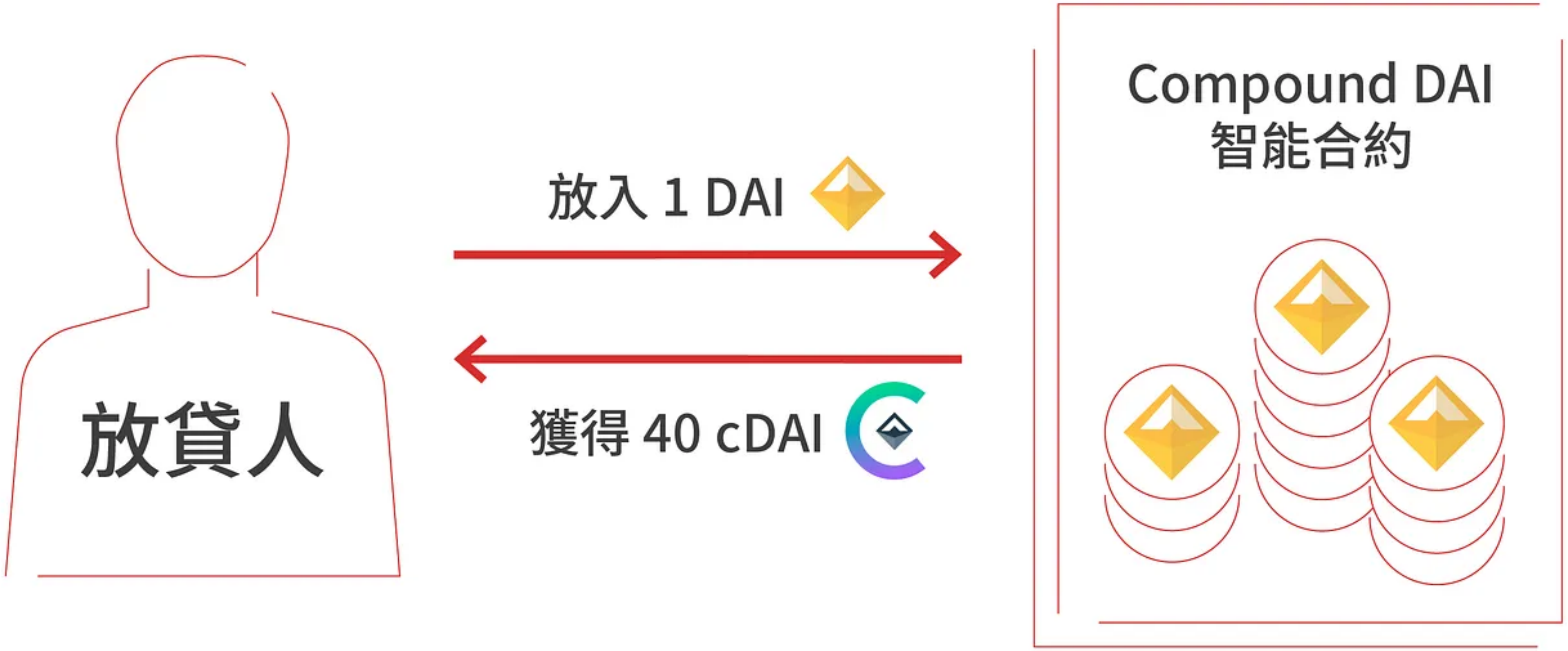
Net Earn APR
2.86%
1.66% Interest
1.19%

Lenders

Compound

Deposit ETHs —> Earning ETH interest

```
function supply(address asset, uint amount)
```



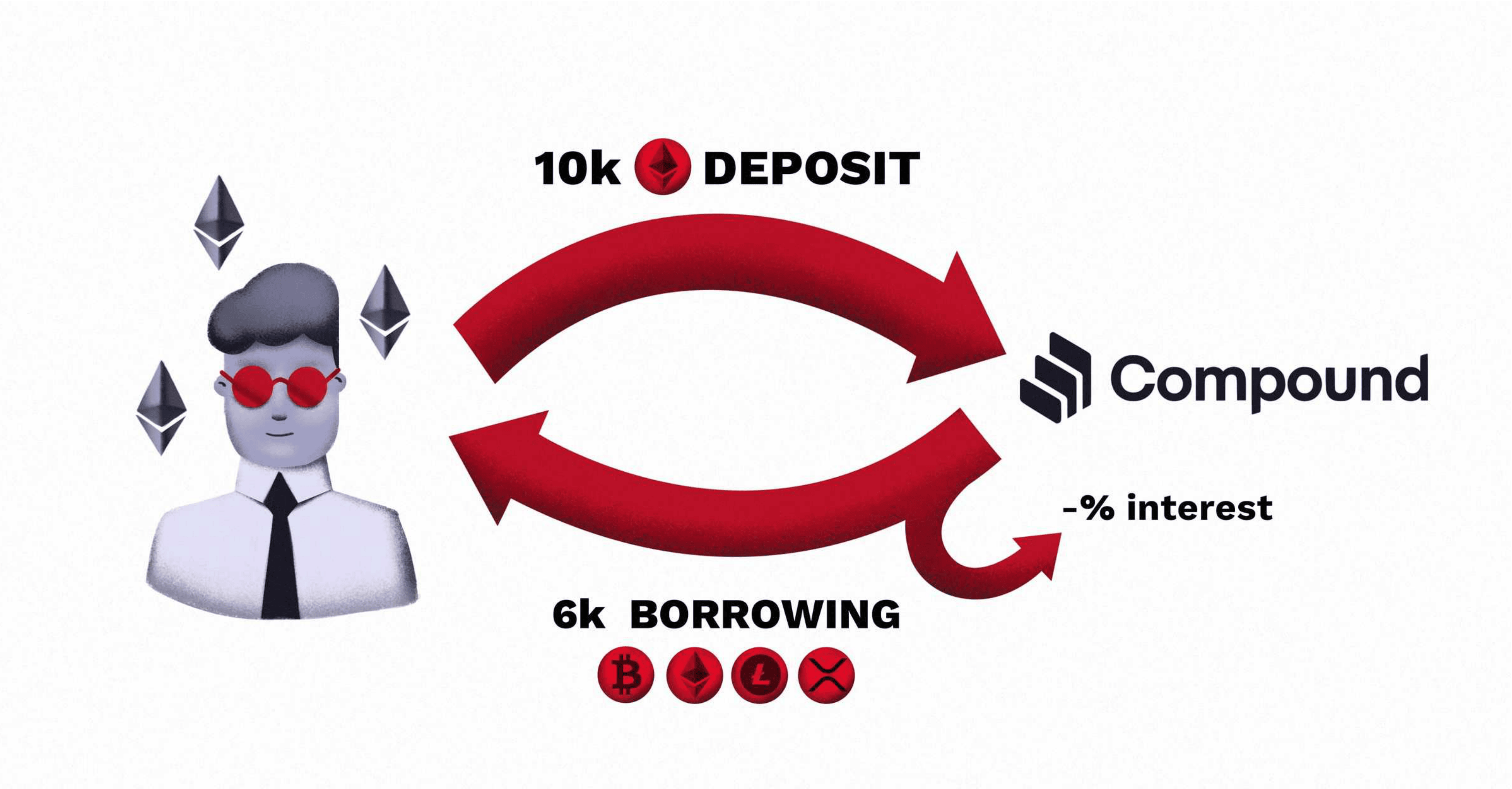
Borrower

Compound

Collateral 10K (ETH,USDT, DAI)asset

—> Borrow 6k WBTC asset

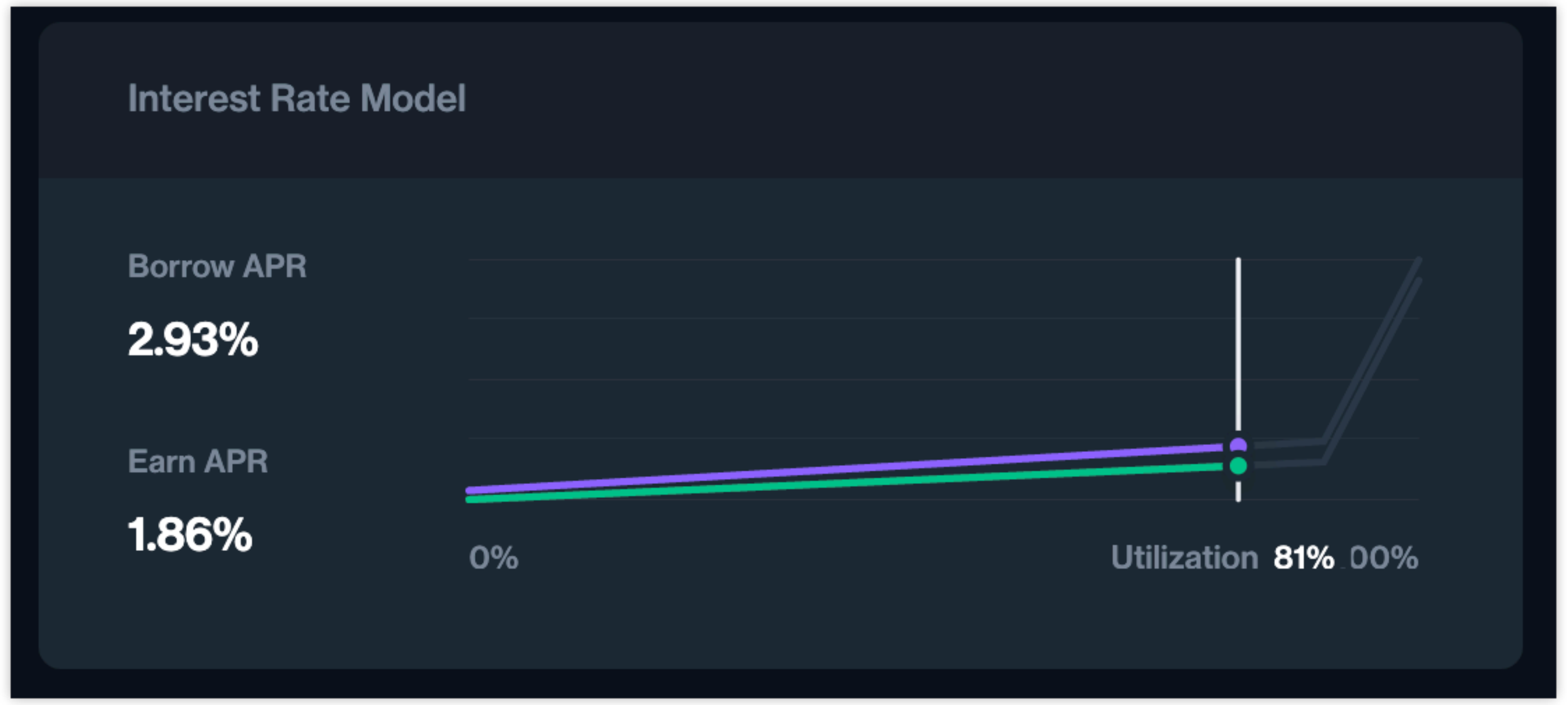
—> pay WBTC interest



How to calculate Interest

Compound

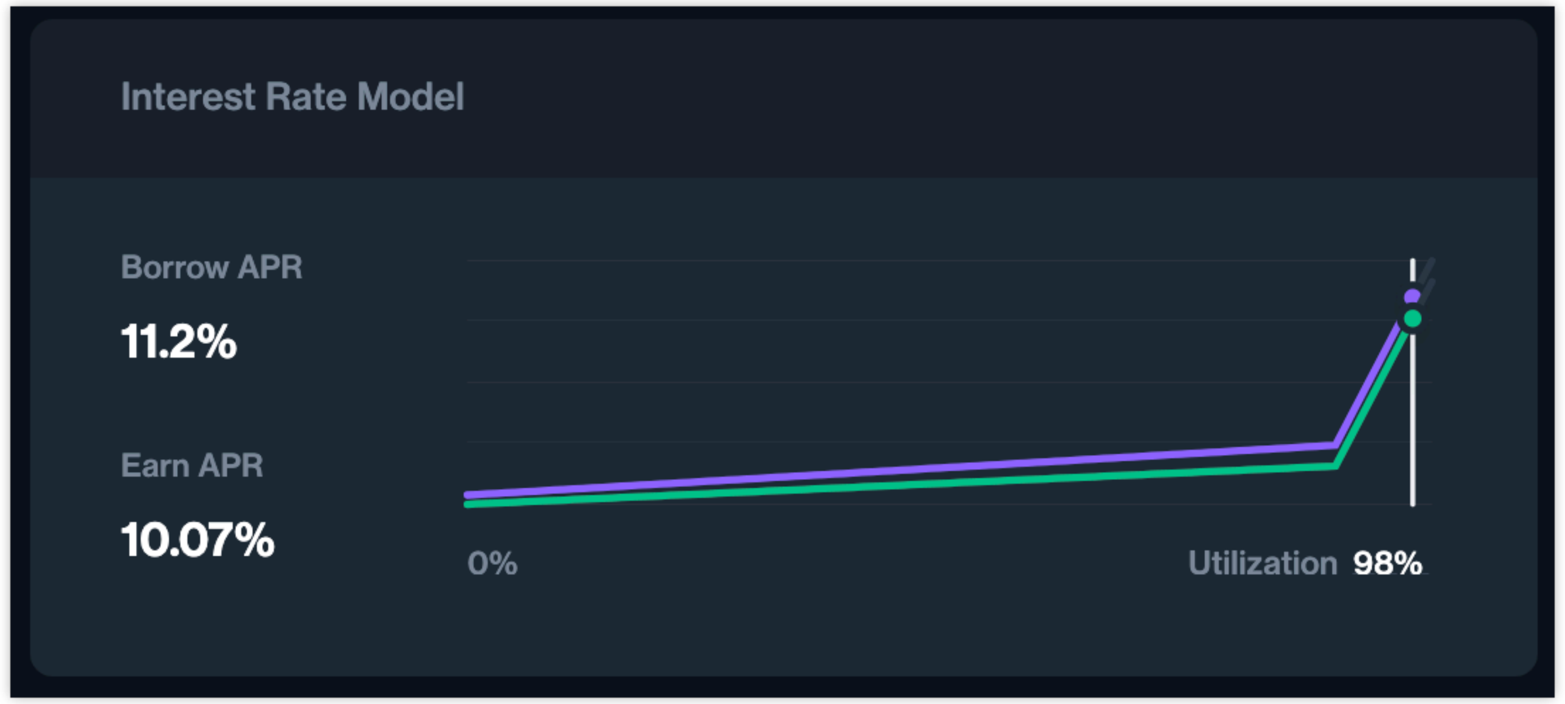
~~interest rates are fixed~~, Compound interest works in a dynamic fashion



Interest Rate Model

Borrow APR
2.93%

Earn APR
1.86%



Interest Rate Model

Borrow APR
11.2%

Earn APR
10.07%

UtilizationRate = borrows / (cash + borrows - reserves)

Borrow Rate = baseRate + utilizationRate* multiplier

Supply Rate = Borrow Rate * utilizationRate * (1 -reserve factor)

GitHub

How to calculate Interest

Compound

UtilizationRate = borrows / (cash + borrows - reserves)

Borrow Rate = baseRate + utilizationRate* multiplier

Supply Rate = Borrow Rate * utilizationRate * (1 -reserveFactor)

Case DAI: const baseRate=5%, multiplier=12%, reserveFactor=5%

When UtilizationRate=0%, Borrow Rate= 5%, Supply Rate=0

When UtilizationRate=100%, Borrow Rate= 17%, Supply Rate=16.15%

How to calculate Interest

Compound

Update interest when Supply、Borrow、Redeem、Repay action

02-12: Alice borrow 1000 DAI , Borrow Rate is 10% at block 1000

02-15: Borrow Rate up to 11%

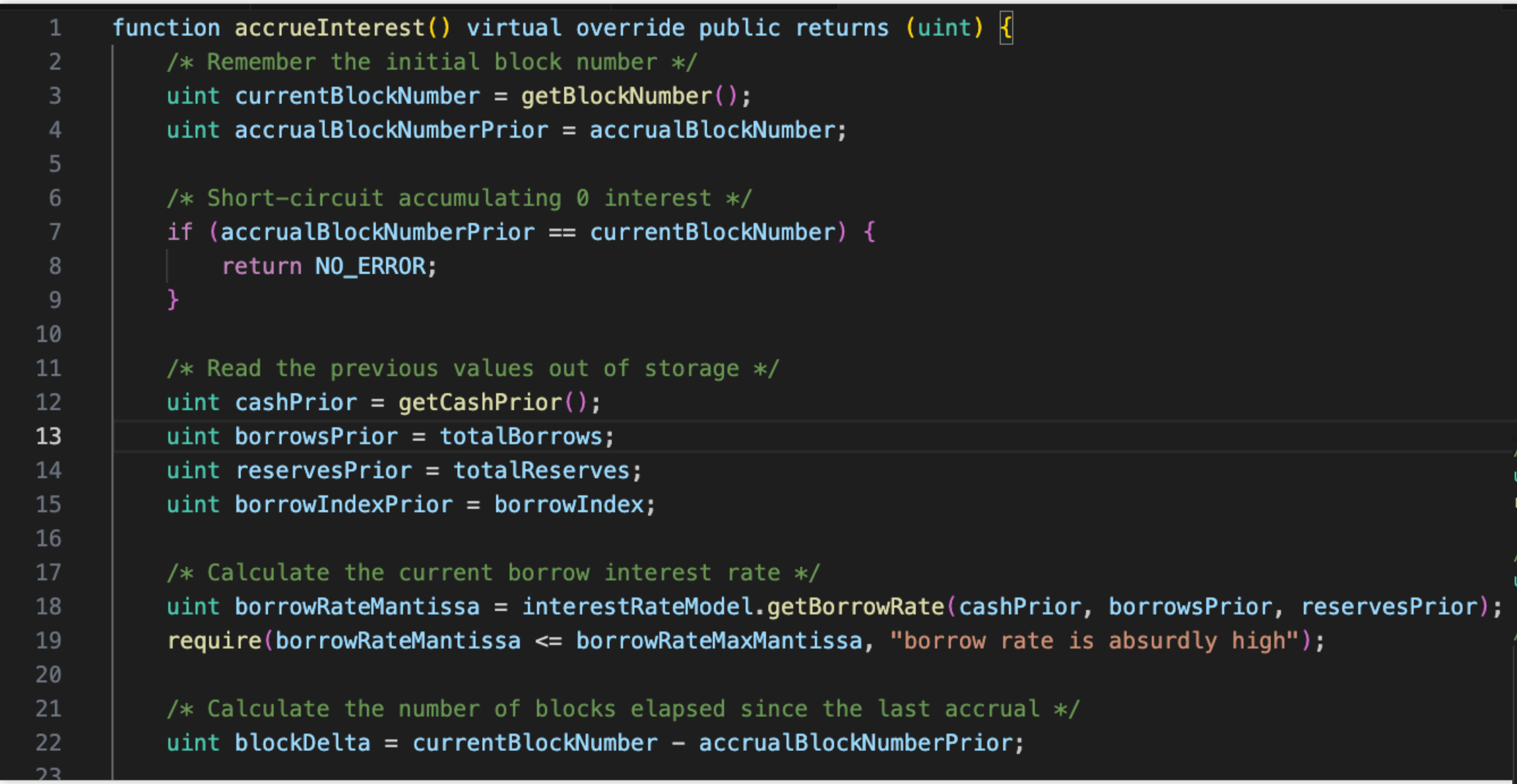
03-19: Alice need pay:

3days(12-15):   1000*10%/365*3 = 0.82191781

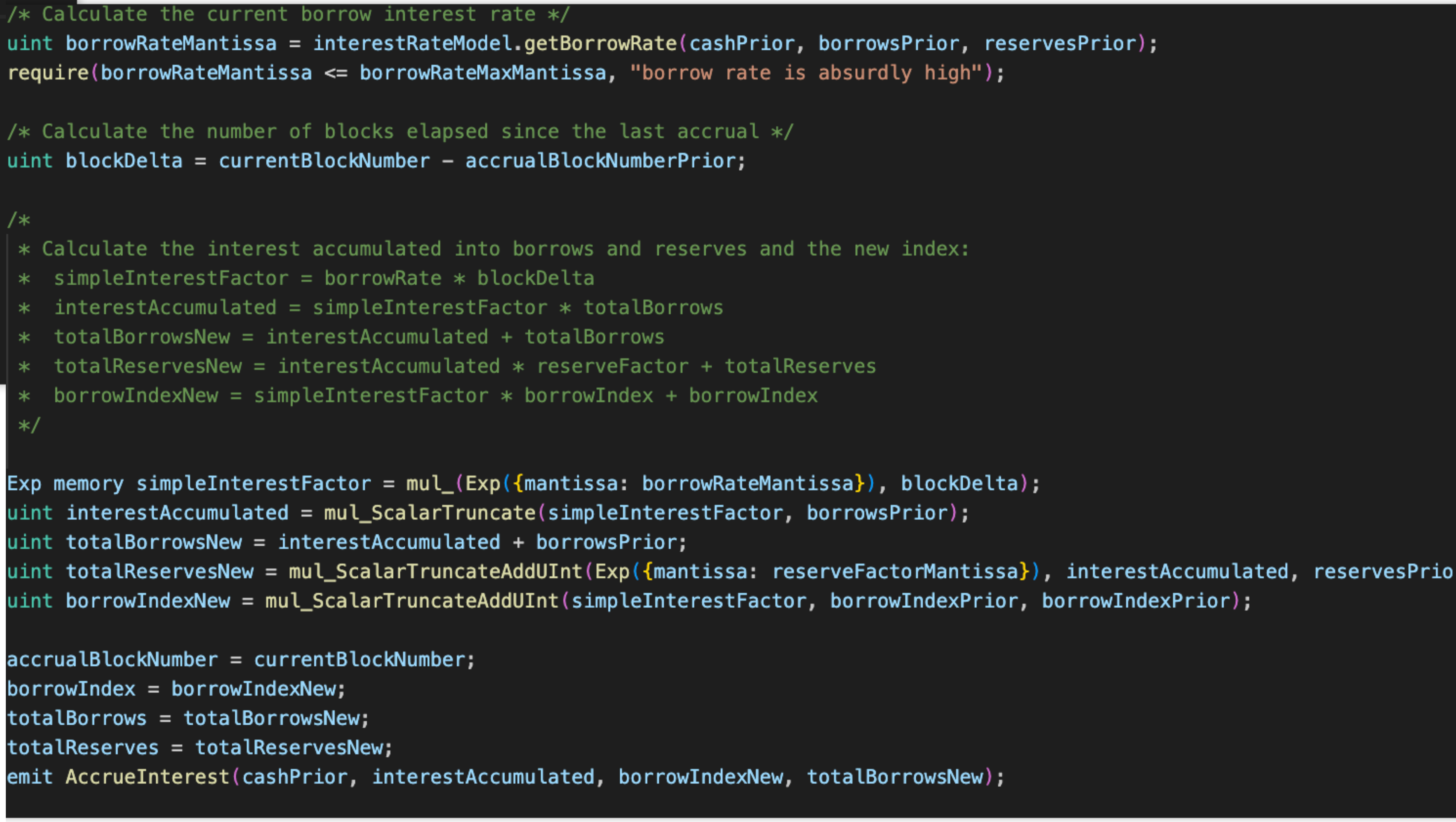
4days(15-19):   (1000+ 0.82191781) * 11%/364*4

How to calculate Interest

Compound



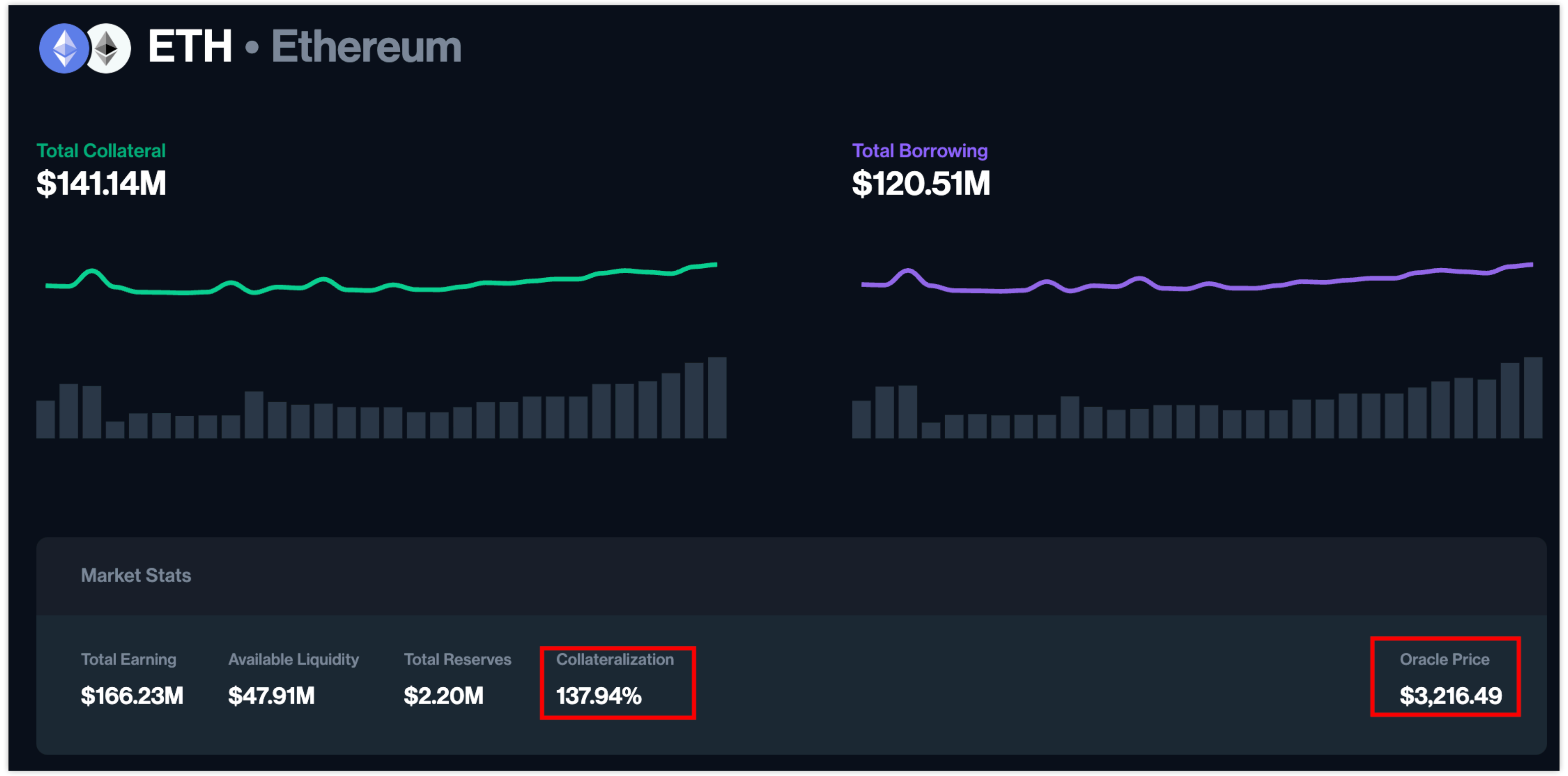
```
function accrueInterest() virtual override public returns (uint) {
    /* Remember the initial block number */
    uint currentBlockNumber = getBlockNumber();
    uint accrualBlockNumberPrior = accrualBlockNumber;

    /* Short-circuit accumulating 0 interest */
    if (accrualBlockNumberPrior == currentBlockNumber) {
        return NO_ERROR;
    }

    /* Read the previous values out of storage */
    uint cashPrior = getCashPrior();
    uint borrowsPrior = totalBorrows;
    uint reservesPrior = totalReserves;
    uint borrowIndexPrior = borrowIndex;

    /* Calculate the current borrow interest rate */
    uint borrowRateMantissa = interestRateModel.getBorrowRate(cashPrior, borrowsPrior, reservesPrior);
    require(borrowRateMantissa <= borrowRateMaxMantissa, "borrow rate is absurdly high");

    /* Calculate the number of blocks elapsed since the last accrual */
    uint blockDelta = currentBlockNumber - accrualBlockNumberPrior;
```



```
/* Calculate the current borrow interest rate */
uint borrowRateMantissa = interestRateModel.getBorrowRate(cashPrior, borrowsPrior, reservesPrior);
require(borrowRateMantissa <= borrowRateMaxMantissa, "borrow rate is absurdly high");

/* Calculate the number of blocks elapsed since the last accrual */
uint blockDelta = currentBlockNumber - accrualBlockNumberPrior;

/*
 * Calculate the interest accumulated into borrows and reserves and the new index:
 *  simpleInterestFactor = borrowRate * blockDelta
 *  interestAccumulated = simpleInterestFactor * totalBorrows
 *  totalBorrowsNew = interestAccumulated + totalBorrows
 *  totalReservesNew = interestAccumulated * reserveFactor + totalReserves
 *  borrowIndexNew = simpleInterestFactor * borrowIndex + borrowIndex
 */

Exp memory simpleInterestFactor = mul_(Exp({mantissa: borrowRateMantissa}), blockDelta);
uint interestAccumulated = mul_ScalarTruncate(simpleInterestFactor, borrowsPrior);
uint totalBorrowsNew = interestAccumulated + borrowsPrior;
uint totalReservesNew = mul_ScalarTruncateAddUInt(Exp({mantissa: reserveFactorMantissa}), interestAccumulated, reservesPrio
uint borrowIndexNew = mul_ScalarTruncateAddUInt(simpleInterestFactor, borrowIndexPrior, borrowIndexPrior);

accrualBlockNumber = currentBlockNumber;
borrowIndex = borrowIndexNew;
totalBorrows = totalBorrowsNew;
totalReserves = totalReservesNew;
emit AccrueInterest(cashPrior, interestAccumulated, borrowIndexNew, totalBorrowsNew);
```

How to borrow

Compound



ETH • Ethereum

Total Collateral
$141.14M

Total Borrowing
$120.51M

Market Stats

Total Earning $166.23M

Available Liquidity $47.91M

Total Reserves $2.20M

Collateralization 137.94%

Oracle Price $3,216.49

Collateral Value / Borrow Value >= Collateralization

Collateral Value = Supply * Asset Oracle Price

Borrowing Value = Borrowing amount * Asset Oracle Price

How to Liquidation

Compound

When Supply $1000 DAI, Borrow $800 ETH:

Rate = $1000/$800 = 125%

After 2 seconds:

When ETH price +5%   Rate = $1000/$840 = 119%

Lower than 120% requires liquidation

Readcode

Compound

Doc: https://docs.compound.finance/interest-rates/

Github:https://github.com/compound-finance/compound-protocol/tree/master

Exercise

- Read : https://learnblockchain.cn/article/5036
- Add NFTMarket Features:
  - 1. Charge NFT transaction fee
  - 2. Support Stake ETH to earn transaction fee