

# Elements of DeFi

<https://web3.princeton.edu/elements-of-defi/>

**Professor** Pramod Viswanath

Princeton University

# **Lecture 12**

## **Flash loans**

# Last Lecture: Lending

- Need for lending in any economy
  - Driven by trust in lenders
  - Track reputation of borrowers
- Decentralized Lending
  - Need for over-collateralization
  - Agents involved and their incentives – lender, borrower, liquidator, oracle
  - Action space – shorting, arbitrage, changing interest rates
- Under-collateralized lending: proposals

# This lecture: Flash loans

- Liquidity pools
  - Lending liquidity pools - Unused capital due to overcollateralization
  - Exchange liquidity pools – Locked collateral for margin traders (dydx) – (Check if AMMs also provide liquidity)
- Trust requirements
  - Replace borrower trust with trust on Ethereum (atomic transaction execution)
  - Replace with incentivized trust (margin trading)
- Flash loans
- Applications:
  - Flash loan arbitrage
  - Flash loan liquidation

# Lending liquidity pool

- Borrowing based on collateral
  - Self sufficient
  - But need capital upfront and capital locked up
- **Peer-to-pool-peer Design Principle**
  - Aggregates capital and distributes
  - Seen previously in AMM design
- Pool formed by collateral deposited by lenders
  - Can be lent to borrowers
- **Key challenge**: how to guarantee borrowers will repay?

# Counterparty Risk

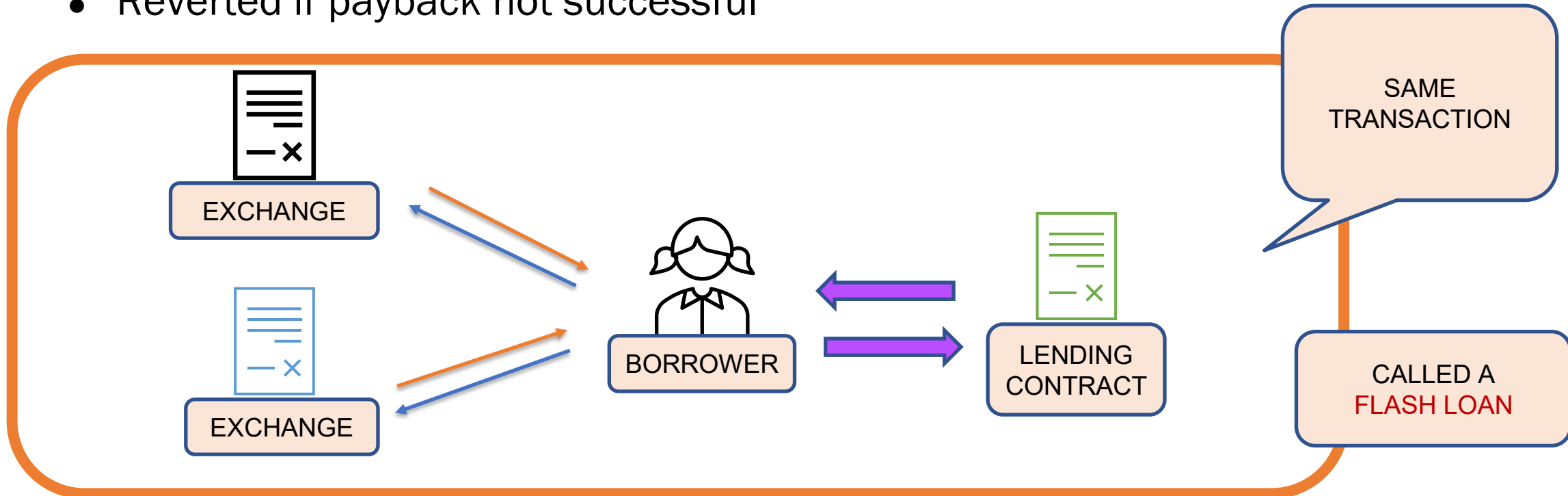
- When lending out tokens, protocol needs to ensure that borrower is incentivized to pay back
- Also needs to ensure that some value can be recovered in case of default = “counterparty risk mitigation”
- Two ways to do this in TradFi –
  - Credit Score / Credit Rating
  - Margin Trading Liquidation

# Counterparty risk mitigation

- **Credit Score / Credit Rating**
  - Typically calculated based on the following metrics
    - Loan or credit repayment history
    - Current outstanding debt
    - Length of credit history
    - Recent debt taken
- **Margin Trading Liquidation**
  - Require the trader to maintain some minimum collateral against borrowed amount
  - Lender reserves the right to sell collateral (**liquidate**) if value goes below threshold

# Counterparty risk mitigation in DeFi

- Blockchains offer us an alternative: **atomicity**
  - Atomic transaction execution
  - Borrow and pay back within the same transaction
  - Reverted if payback not successful





# Flash loans

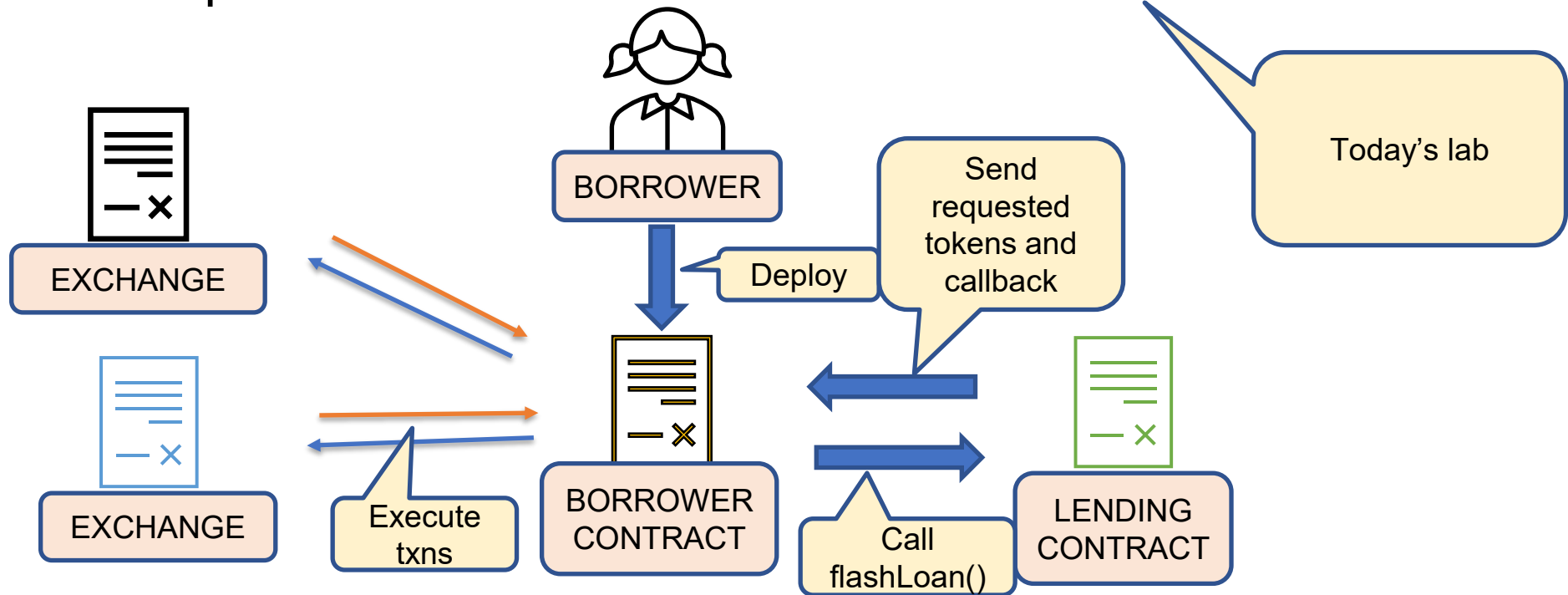
- Introduced by the Marble lending protocol in 2018
- Popularized by Aave in 2020 –use took off after bZx attack in 02/2020

## Features

- Atomicity implies that lender's funds are not at risk
- Thus, no collateral or interest needs to be posted!
- Borrower pays a small fee ( $\sim 0.1\%$ ) to lenders who contributed to pool

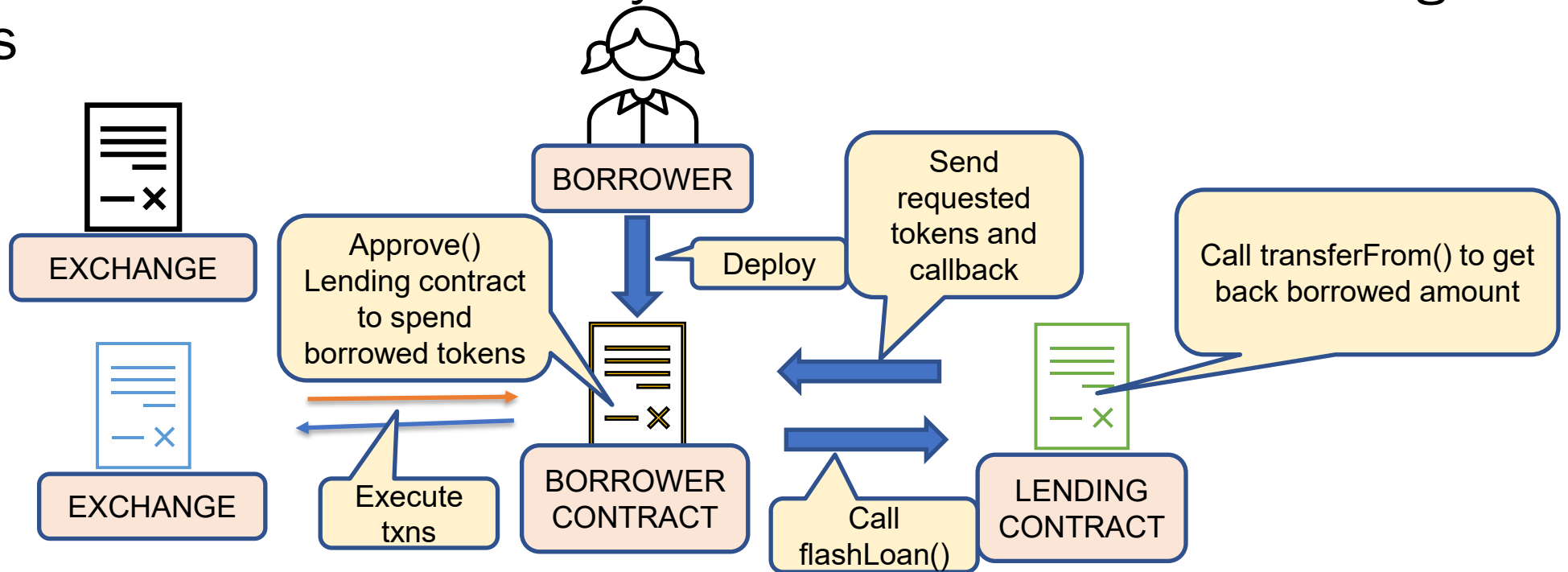
# Flash loans - implementation

- Flash loans are implemented as part of the main lending contract of protocols like Aave
- The function executing a flash loan uses the callback feature of EVM that calls a specific function on the borrower contract



# Flash loans - implementation

- After callback completes all of its actions, it approves the lending contract to get its borrowed tokens
- Execution resumes in the flashLoan() function of the lending contract
- Function executes successfully if borrower contract has enough funds

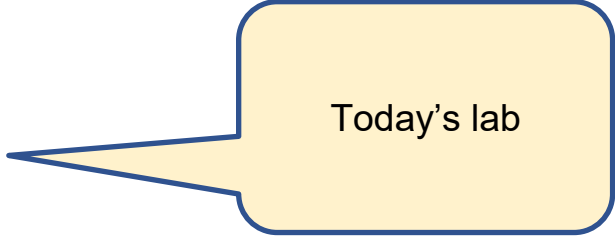


# Flash loans - incentives

## Fee mechanisms

- Fees to incentivize lenders charge  $\sim 0.1\%$
- That is, borrower pays back  $0.1\%$  extra in the borrowed token
- Lenders also get back tokens risk-free

# Flash loan based arbitrage



Today's lab

- If a trader spots an arbitrage opportunity, they can immediately exploit it using a flash loan
- No capital needed and the profit is riskless'
- Information can be converted to money directly without any capital
- However, because other traders might also spot the same opportunity, gas price bidding might happen

# Flash loan based arbitrage

## Example arbitrage transaction -

From: [0x8645abffe4fad9e0c6c18afff30ef6aea438008c](#)

To: [Contract 0x398ec7346dcd622edc5ae82352f02be94c62d119](#)

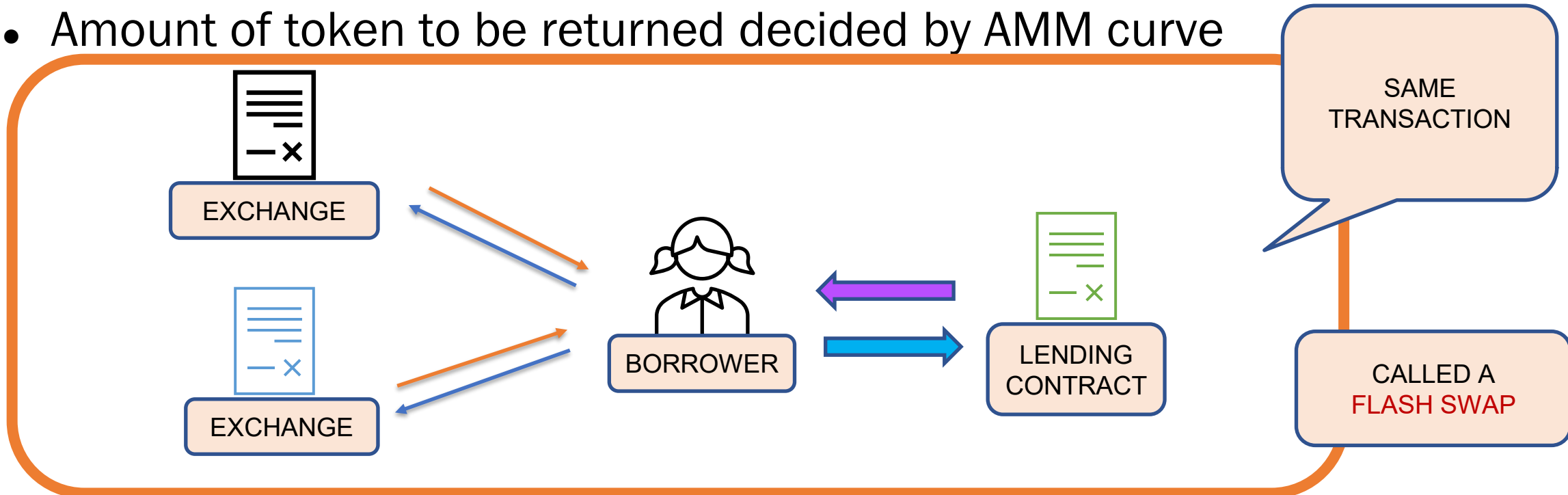
Tokens Transferred:  
(8 ERC-20 Transfers found)

- TRANSFER 18.162114996348982672 Ether From [0x09cabec1ead1c0ba254...](#) To [0x594f7f38c97a1847b600...](#)
- TRANSFER 18.162114996348982672 Ether From [0x594f7f38c97a1847b600...](#) To [0x2a1530c4c41db0b2...](#)
- From [0x3dfd23a6c5e8bb...](#) To [0x594f7f38c97a184...](#) For 3,137.407014296228788899 (\$3,124.54) Dai Stableco... (DAI)
- From [0x594f7f38c97a184...](#) To [0xc73e0383f3aff321...](#) For 3,137.407014296228788899 (\$3,124.54) Dai Stableco... (DAI)
- From [0xc73e0383f3aff321...](#) To [0x0000000000000000...](#) For 3,137.407014296228788899 (\$3,124.54) Dai Stableco... (DAI)
- From [0xad37fd42185ba6...](#) To [0x594f7f38c97a184...](#) For 3,137.407014296228788899 (\$3,186.31) Sai Stableco... (SAI)
- From [0x594f7f38c97a184...](#) To [0x09cabec1ead1c0...](#) For 3,137.407014296228788899 (\$3,186.31) Sai Stableco... (SAI)
- From [0x2a1530c4c41db0...](#) To [0x594f7f38c97a184...](#) For 3,157.412913230790346677 (\$3,144.46) Dai St...
- From [0x594f7f38c97a184...](#) To [0x3dfd23a6c5e8bb...](#) For 3,148.38793884626558966 (\$3,135.47) Dai St...
- From [0x3dfd23a6c5e8bb...](#) To [0xffac71f2395fd2a4...](#) For 3.294277365011040228 (\$3.28) Dai Stableco...

Profits are modest per transaction, but can exploit many such opportunities without any cost

# Flash swaps

- Exchanges (e.g., Uniswap) have incorporated flash loans: **flash swaps**
- Same as flashloans – but token returned is different from token borrowed
- Amount of token to be returned decided by AMM curve



# Flash loans for swapping collateral

1. A flash loan is taken
  2. Loan is used to repay debt and withdraw collateral
  3. Swap old collateral with new collateral
  4. Use new collateral to take out the same loan
  5. Repay the flash loan
- **Modify short positions**

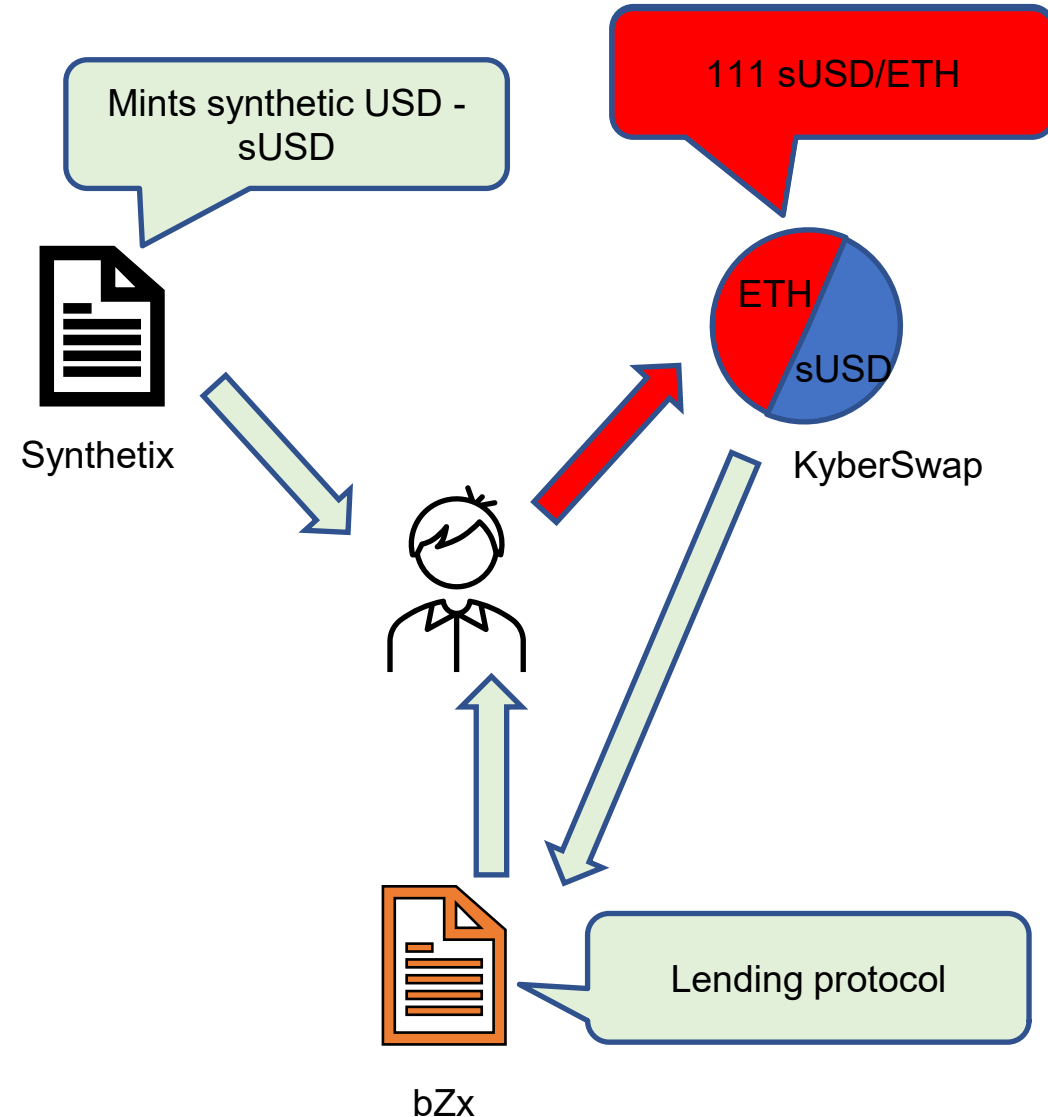


# Flash loans for Self-Liquidation

- Recall: liquidation happened when value of collateral fell causing the value of borrowed token to be above the liquidation threshold
- In such a case, a liquidator pays back part of the loans to get rewarded
- Borrower penalized a lot (~5-10% of collateral)
- Instead, borrower takes a flash loan, repays the original loan and gets collateral for almost no fee

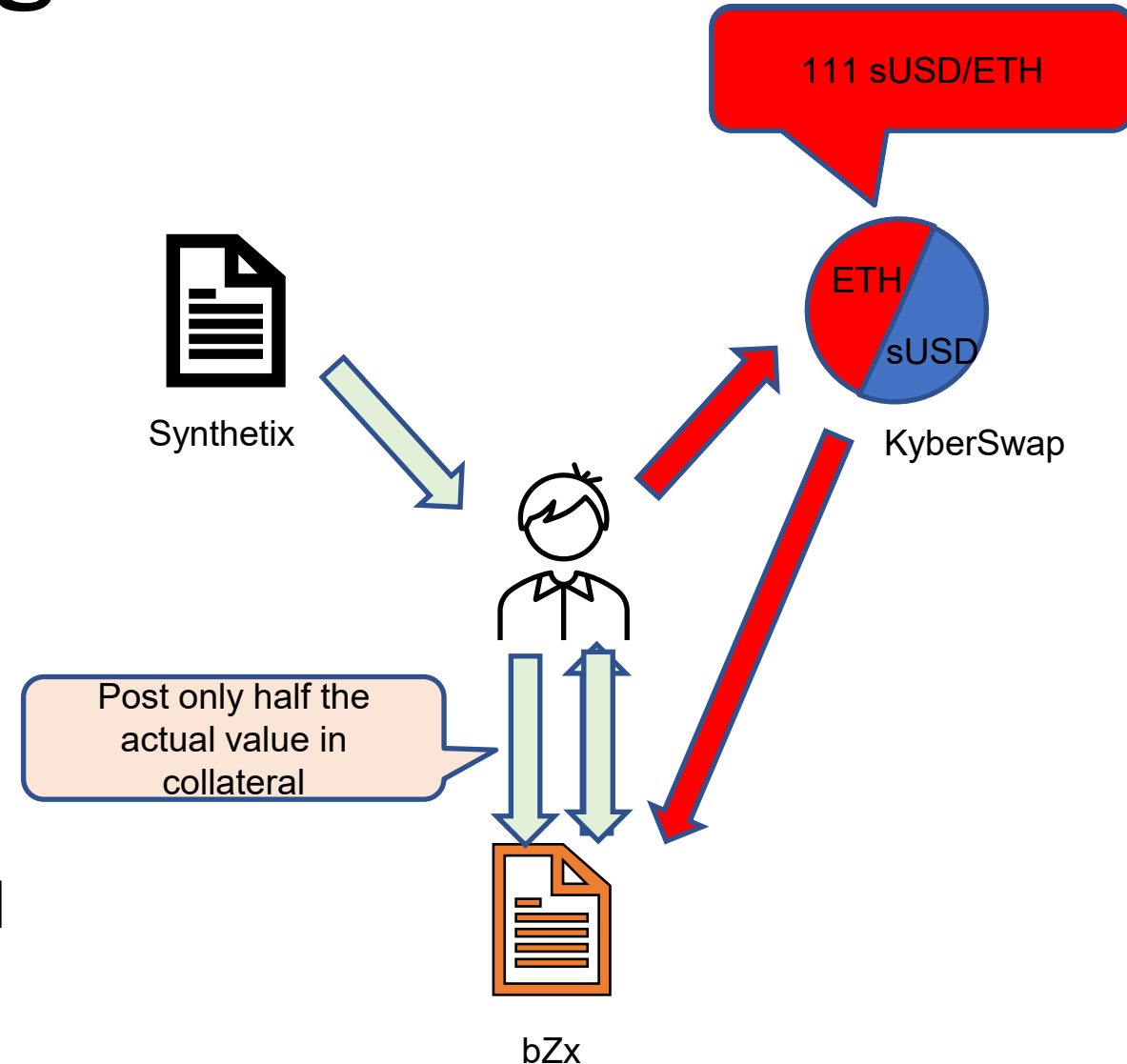
# Recall: bZx attack using FlashLoan

- Step 1 : Borrow 7500 ETH from bZX – promise to repay in same block (flash loan)
- Step 2 : Sell 900 ETH on Kyber pool – changes price by a lot
- Step 3 : Get 943k sUSD for 3518 ETH on Synthetix



# Recall: bZx attack using FlashLoan

- Step 4 : Borrow 6796 ETH from bZX – post collateral which is priced through Kyber – 1099k sUSD
- Step 5 : Repay 7500 ETH flash loan
- Step 6 : Run away with 2378 ETH profit!



# Conclusion

- Minimal capital requirement
  - Balance market and maintain prices
  - Efficient financial ecosystem
- Key challenge
  - Increased attack vector space
  - Adversaries can also act efficiently