Agenda

- Introduction To Blockchain
- Introduction To DeFi
- Introduction to Flash Loan
- The Blessings of Flash Loan
- The Cures of Flash Loan
- How to Prevent Flash Loan Attacks

#HITB2024BKK

Introduction To Blockchain

#HITB2024BKK

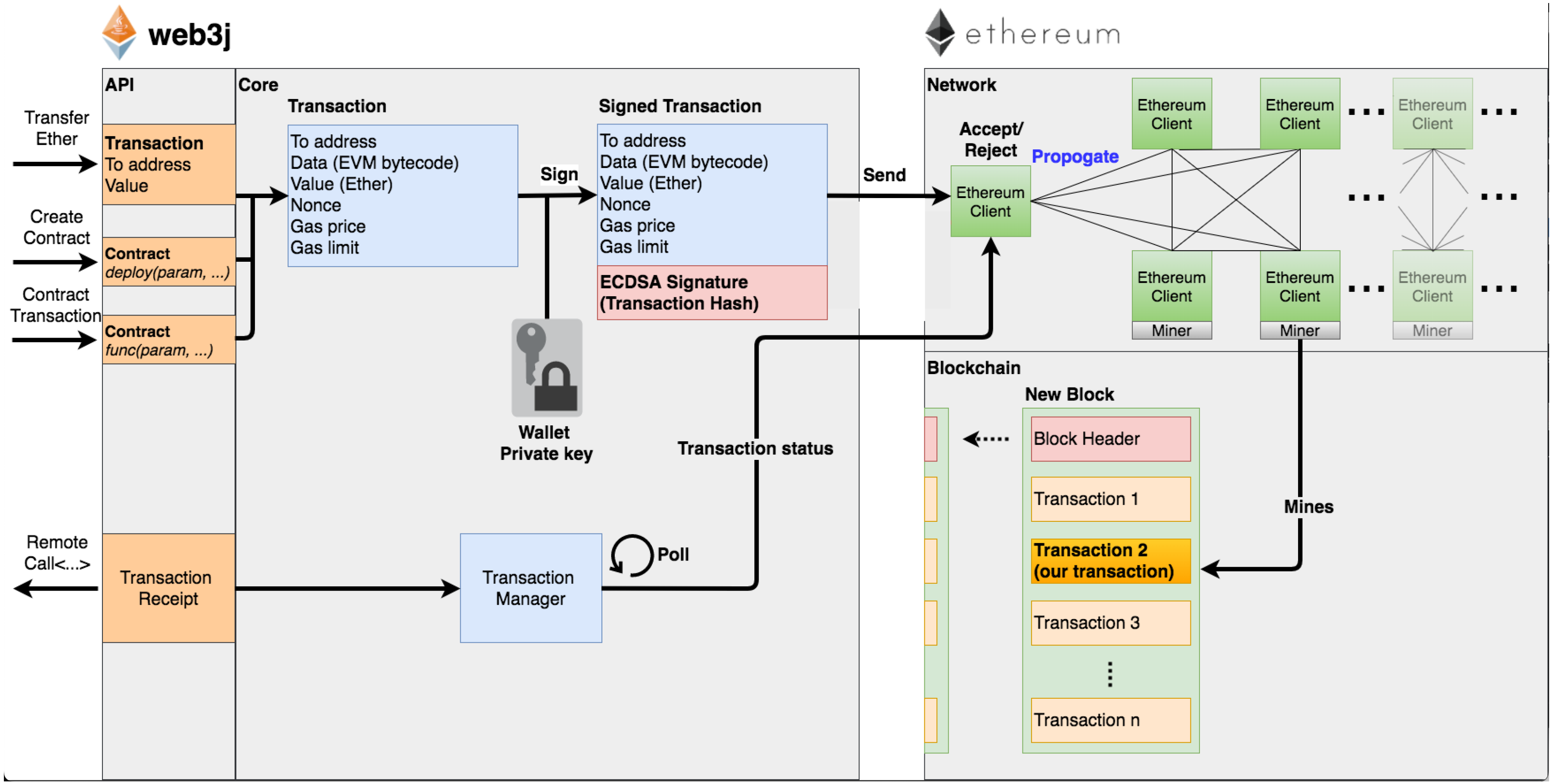
Blockchain

THE 3 WORDS

Smat Contract:
Self-executing contracts with the terms of the agreement directly written into code on the blockchain. In the Ethereum network, smart contracts automatically execute and enforce transactions or actions when predetermined conditions are met. They are transparent, secure, and tamper-proof, eliminating the need for intermediaries and ensuring trust between parties.

Transaction:
An action on the Ethereum blockchain, such as transferring Ether (ETH) or deploying/executing a smart contract. Each transaction is verified by network participants and recorded in a block, which is then added to the blockchain. These transactions are irreversible and transparent.

Gas:
A fee paid to process transactions on the blockchain, particularly in networks like Ethereum. Gas ensures that users compensate validators for the computational work required to validate and execute transactions.



web3j

ethereum

API

Transfer Ether

Transaction
To address
Value

Create Contract

Contract
deploy(param, ...)

Contract Transaction

Contract
func(param, ...)

Remote Call<...>

Transaction Receipt

Core

Transaction
To address
Data (EVM bytecode)
Value (Ether)
Nonce
Gas price
Gas limit

Sign

Signed Transaction
To address
Data (EVM bytecode)
Value (Ether)
Nonce
Gas price
Gas limit
ECDSA Signature (Transaction Hash)

Wallet
Private key

Transaction status

Transaction Manager

Poll

Send

Network

Accept/ Reject

Propogate

Ethereum Client

Ethereum Client

Ethereum Client

Ethereum Client

Ethereum Client
Miner

Ethereum Client
Miner

Ethereum Client
Miner

Blockchain

New Block

Block Header

Transaction 1

Transaction 2 (our transaction)

Transaction 3

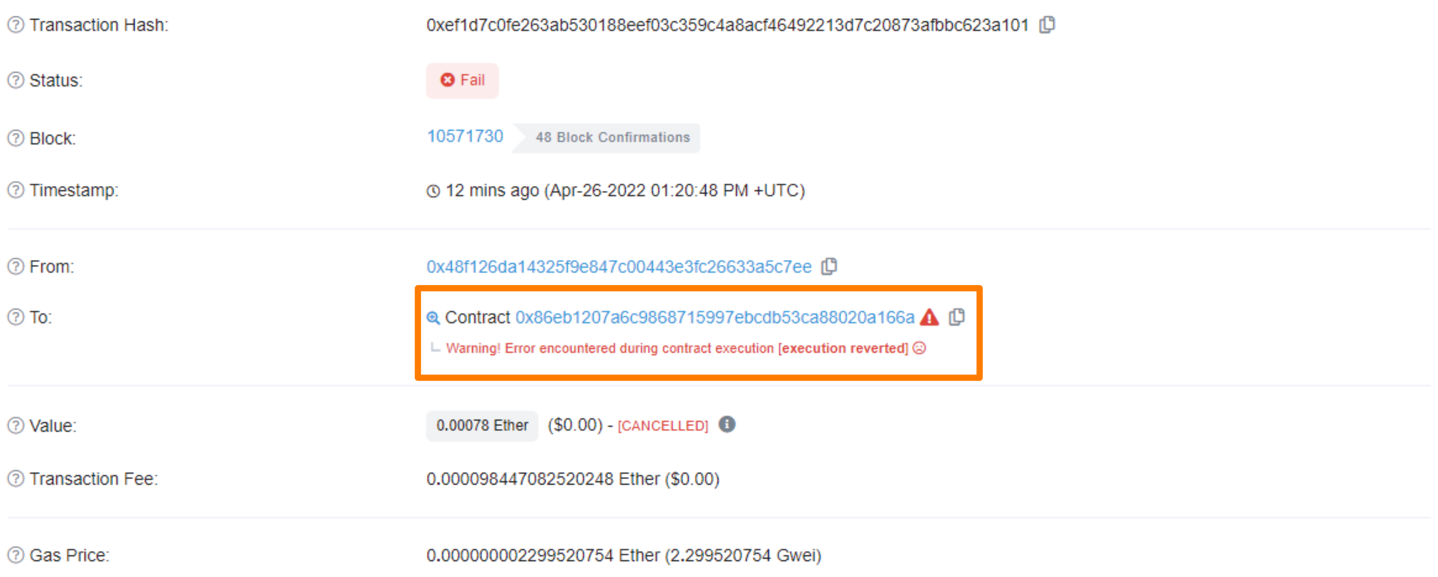
Transaction n

Mines

Ref: https://docs.web3j.io/4.11.0/transactions/transactions/

#HITB2024BKK

Reverting due to unmet conditions

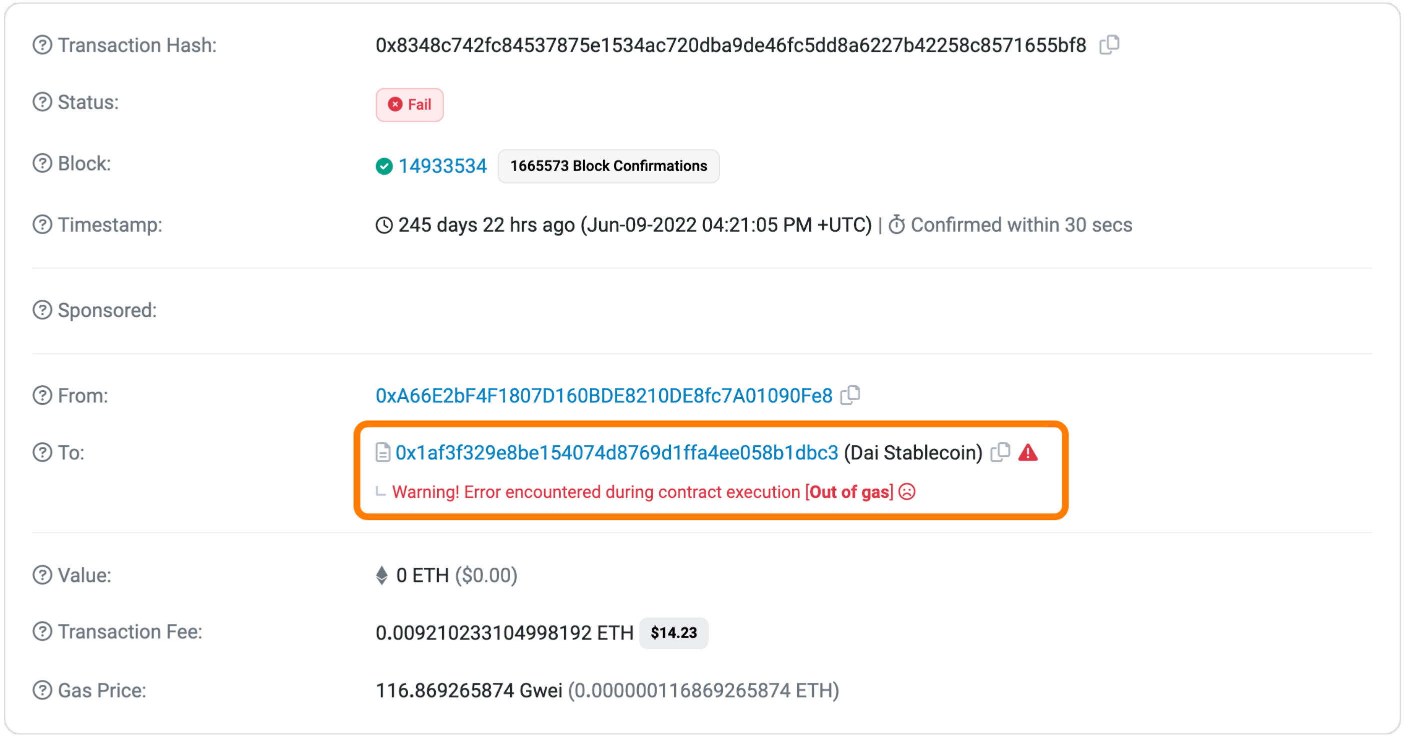


Transaction Hash:	0xef1d7c0fe263ab530188eef03c359c4a8acf46492213d7c20873afbbc623a101
Status:	Fail
Block:	10571730 48 Block Confirmations
Timestamp:	12 mins ago (Apr-26-2022 01:20:48 PM +UTC)
From:	0x48f126da14325f9e847c00443e3fc26633a5c7ee
To:	Contract 0x86eb1207a6c9868715997ebcdb53ca88020a166a ⚠ Warning! Error encountered during contract execution **[execution reverted]** ☹
Value:	0.00078 Ether ($0.00) - [CANCELLED]
Transaction Fee:	0.000098447082520248 Ether ($0.00)
Gas Price:	0.000000002299520754 Ether (2.299520754 Gwei)

Ref: https://support.metamask.io/transactions-and-gas/gas-fees/why-did-my-transaction-fail-with-an-out-of-gas-error-how-can-i-fix-it/

#HITB2024BKK

Reverting due to insufficient gas



Transaction Hash:	0x8348c742fc84537875e1534ac720dba9de46fc5dd8a6227b42258c8571655bf8
Status:	Fail
Block:	14933534 1665573 Block Confirmations
Timestamp:	245 days 22 hrs ago (Jun-09-2022 04:21:05 PM +UTC) | Confirmed within 30 secs
Sponsored:	
From:	0xA66E2bF4F1807D160BDE8210DE8fc7A01090Fe8
To:	0x1af3f329e8be154074d8769d1ffa4ee058b1dbc3 (Dai Stablecoin) Warning! Error encountered during contract execution **[Out of gas]** ☹
Value:	0 ETH ($0.00)
Transaction Fee:	0.009210233104998192 ETH $14.23
Gas Price:	116.869265874 Gwei (0.000000116869265874 ETH)

Ref: https://stackoverflow.com/questions/72015057/transaction-failed-with-execution-error-while-sending-ether

#HITB2024BKK

Introduction To DeFi

Decentralized Finance (DeFi)

Financial ecosystem built on blockchain technology, that eliminates intermediaries like humans. DeFi enables users to access financial services such as lending, borrowing, trading, and earning interest through smart contracts, which are **automated** and **self-executing**. This open and permissionless system allows anyone with an internet connection to participate, **offering greater transparency, security, and accessibility**.

Type of DeFi

Concept from Traditional Finance (with Blockchain Variations)
- Lending and Borrowing (but over collateral)
- Stablecoin
- Futures/Options Trading
- Insurance
- etc.

Unique to Blockchain:
- Decentralized Exchanges (DEXs)
- Yield Farming
- Tokenization of Real World Assets (RWA)
- Decentralized Identity
- etc.

Price Feed

Price feeds provide real-time, accurate market data for assets on the blockchain. In DeFi, like lending, borrowing. Accurate price feeds help determine collateral values, trigger liquidations, and maintain stability, ensuring that DeFi protocols operate securely and efficiently.

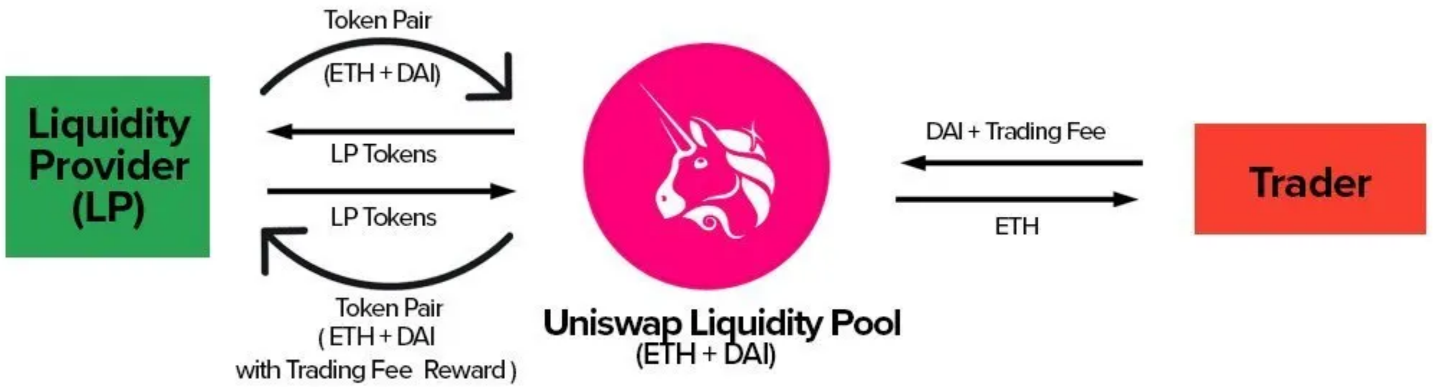
Off-Chain Price Feed: price data is sourced from external markets and brought onto the blockchain through oracles.

On-Chain Price Feed: price data is sourced and maintained entirely within the blockchain network.

On-Chain Price With DEX

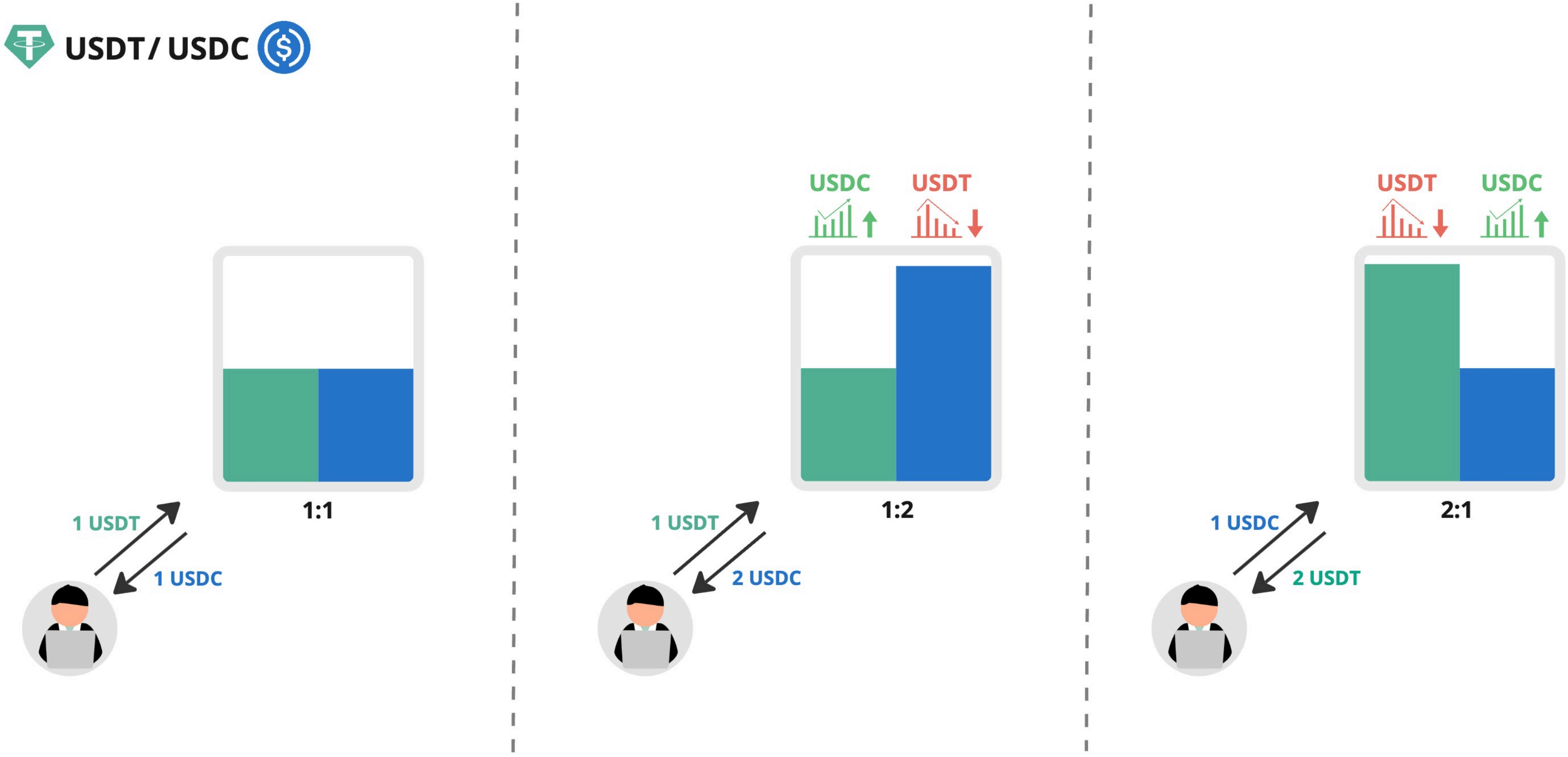
Decentralized Exchanges (DEXs) determine the price between two tokens using automated market maker (AMM) algorithms. In AMMs, users provide liquidity to a pool containing pairs of tokens. **The price between the two tokens is calculated based on the ratio of the tokens in the pool**. **As trades occur, this ratio changes, automatically adjusting the price according to supply and demand**. This mechanism ensures continuous price discovery without relying on the On-Chain Price Feed.

DUAL ASSET LIQUIDITY POOL



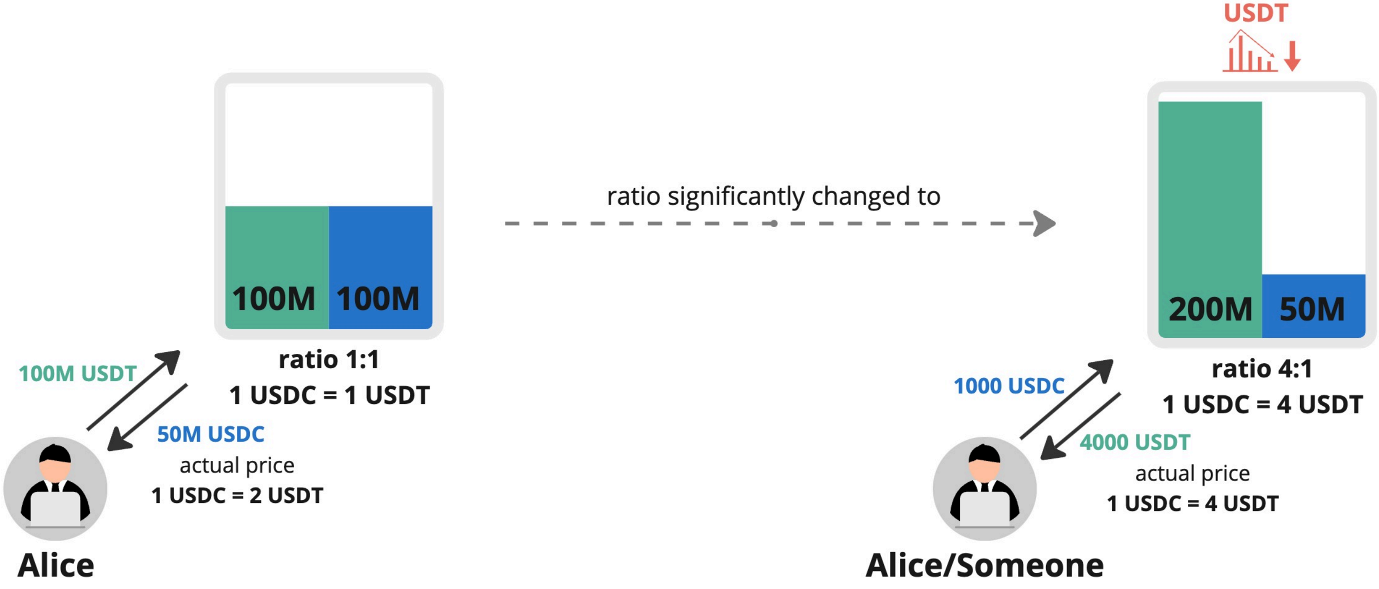
Ref: https://coinsutra.com/liquidity-pools-guide/

USDT / USDC



#HITB2024BKK

USDT / USDC



#HITB2024BKK

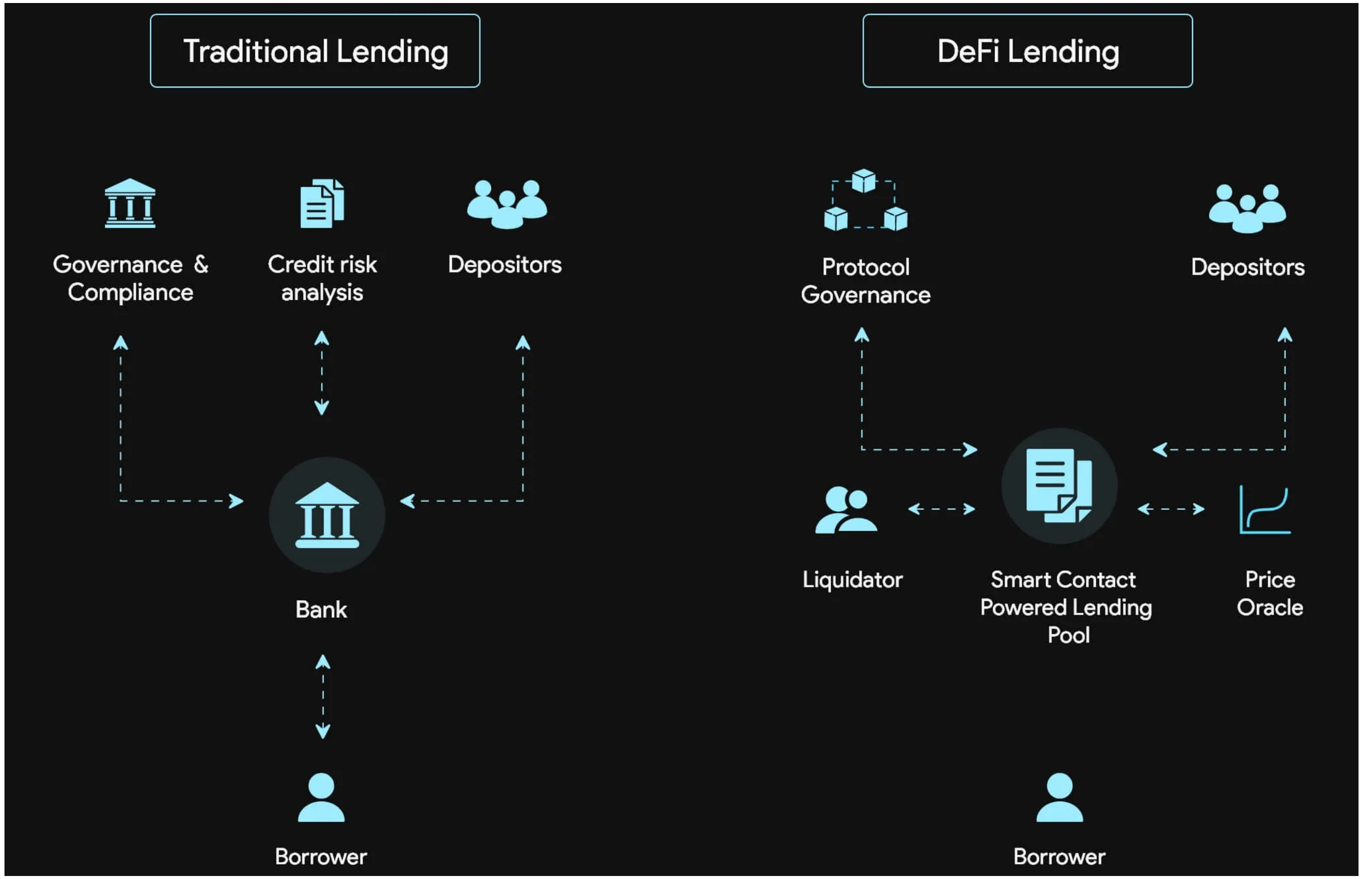
Introduction to Flash Loan

#HITB2024BKK

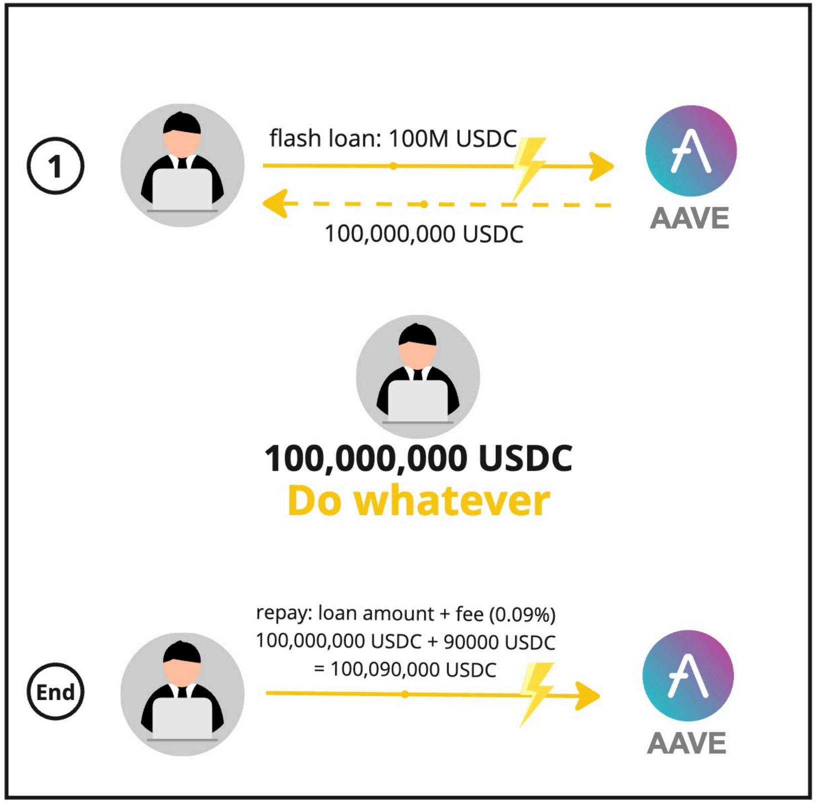
> “for the first time ever, you don't need money to make more money.”
>
> Santiago Palladino, Aztec Network

Ref: https://x.com/smpalladino/status/1230233789311471618

#HITB2024BKK



Ref: https://www.apptunix.com/blog/explore-how-defi-lending-works/



1

flash loan: 100M USDC

100,000,000 USDC

AAVE

100,000,000 USDC

Do whatever

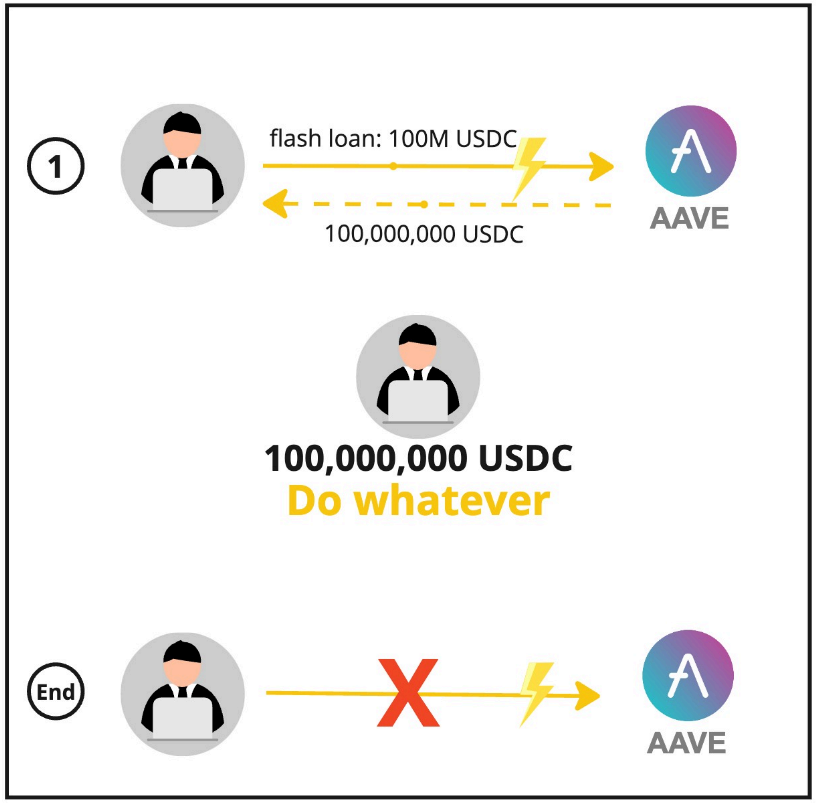
End

repay: loan amount + fee (0.09%)
100,000,000 USDC + 90000 USDC
= 100,090,000 USDC

AAVE

Tx will succeed by paying only for gas.

OR



1

flash loan: 100M USDC

100,000,000 USDC

AAVE

100,000,000 USDC

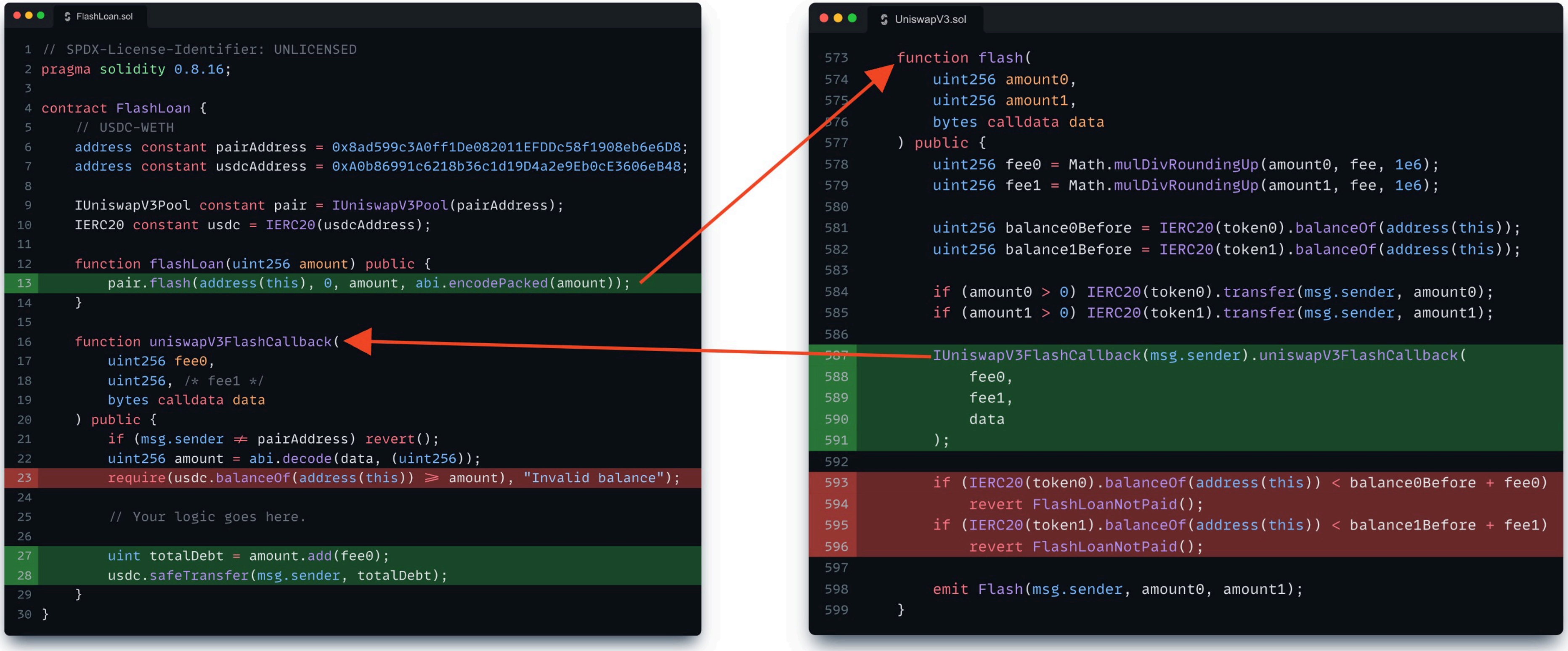
Do whatever

End

AAVE

Tx will reverted but still have to pay for gas.

#HITB2024BKK



```solidity
// SPDX-License-Identifier: UNLICENSED
pragma solidity 0.8.16;

contract FlashLoan {
    // USDC-WETH
    address constant pairAddress = 0x8ad599c3A0ff1De082011EFDDc58f1908eb6e6D8;
    address constant usdcAddress = 0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48;

    IUniswapV3Pool constant pair = IUniswapV3Pool(pairAddress);
    IERC20 constant usdc = IERC20(usdcAddress);

    function flashLoan(uint256 amount) public {
        pair.flash(address(this), 0, amount, abi.encodePacked(amount));
    }

    function uniswapV3FlashCallback(
        uint256 fee0,
        uint256, /* fee1 */
        bytes calldata data
    ) public {
        if (msg.sender != pairAddress) revert();
        uint256 amount = abi.decode(data, (uint256));
        require(usdc.balanceOf(address(this)) >= amount), "Invalid balance");

        // Your logic goes here.

        uint totalDebt = amount.add(fee0);
        usdc.safeTransfer(msg.sender, totalDebt);
    }
}
```

```solidity
    function flash(
        uint256 amount0,
        uint256 amount1,
        bytes calldata data
    ) public {
        uint256 fee0 = Math.mulDivRoundingUp(amount0, fee, 1e6);
        uint256 fee1 = Math.mulDivRoundingUp(amount1, fee, 1e6);

        uint256 balance0Before = IERC20(token0).balanceOf(address(this));
        uint256 balance1Before = IERC20(token1).balanceOf(address(this));

        if (amount0 > 0) IERC20(token0).transfer(msg.sender, amount0);
        if (amount1 > 0) IERC20(token1).transfer(msg.sender, amount1);

        IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(
            fee0,
            fee1,
            data
        );

        if (IERC20(token0).balanceOf(address(this)) < balance0Before + fee0)
            revert FlashLoanNotPaid();
        if (IERC20(token1).balanceOf(address(this)) < balance1Before + fee1)
            revert FlashLoanNotPaid();

        emit Flash(msg.sender, amount0, amount1);
    }
```

#HITB2024BKK

Use Cases of Flash Loan

1. Arbitrage
2. Collateral Swap
3. Self Liquidation
4. Market Making

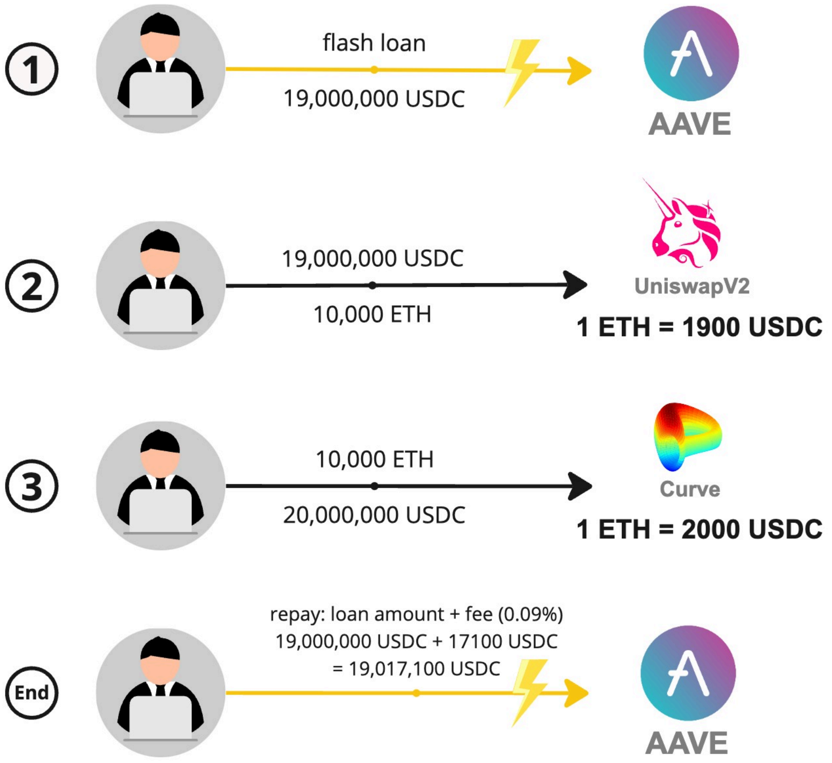
Arbitrage

UniswapV2

1 ETH = 1900 USDC

Curve

1 ETH = 2000 USDC



① flash loan 19,000,000 USDC AAVE

② 19,000,000 USDC 10,000 ETH UniswapV2 **1 ETH = 1900 USDC**

③ 10,000 ETH 20,000,000 USDC Curve **1 ETH = 2000 USDC**

End repay: loan amount + fee (0.09%) 19,000,000 USDC + 17100 USDC = 19,017,100 USDC AAVE

Take profit : 982,900 USDC

Well-known flash loan protocols

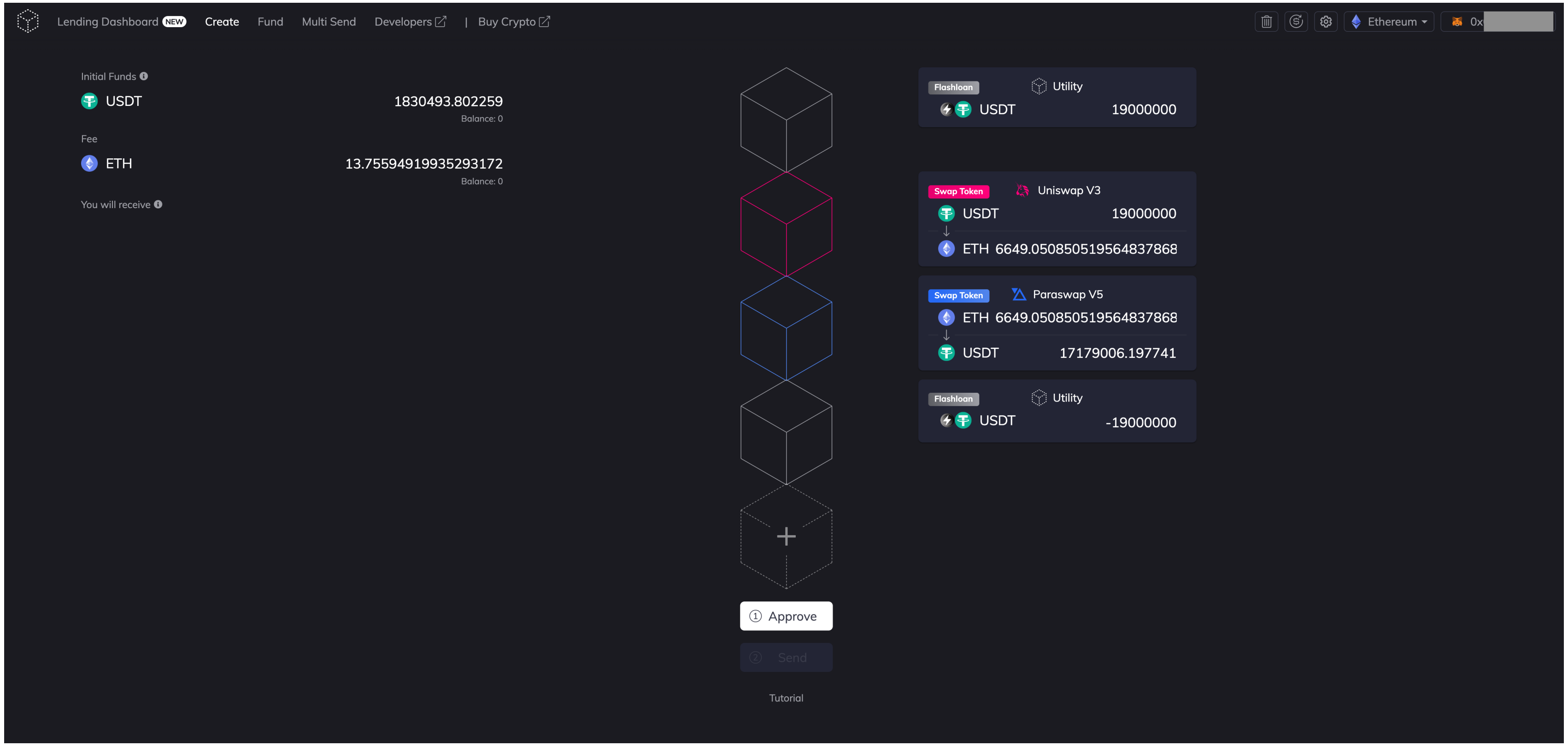
- Aave (Lending/Borrowing Protocol)
- Uniswap (DEX)
- Balancer (DEX)
- dYdX (DEX)
- MakerDAO (Stablecoin/Credit Protocol)

The Blessings of Flash Loan

Blessings for Traders

1. Access to Large Capital Without Collateral
2. Profit from Arbitrage
3. Debt Refinancing and Position Management
4. Enabling Complex Strategies
5. Low-Cost Operations
6. There are programming and non-programming tools.

Non-programming tool



Initial Funds

USDT 1830493.802259
Balance: 0

Fee

ETH 13.75594919935293172
Balance: 0

You will receive

Flashloan — Utility
USDT 19000000

Swap Token — Uniswap V3
USDT 19000000
↓
ETH 6649.050850519564837868

Swap Token — Paraswap V5
ETH 6649.050850519564837868
↓
USDT 17179006.197741

Flashloan — Utility
USDT -19000000

#HITB2024BKK

Programming Tools (Foundry/Hardhat)

Smart Contract Developer

- Develop
- Test
- Test (with Fork)
- Deploy

Smart Contract Auditor

- Create PoC of Attack

Trader

- Fork and try strategies

Hacker/Attacker

- Fork and try strategies

Foundry

[Fork](#)

[Cheat codes](#)

[Example](#)

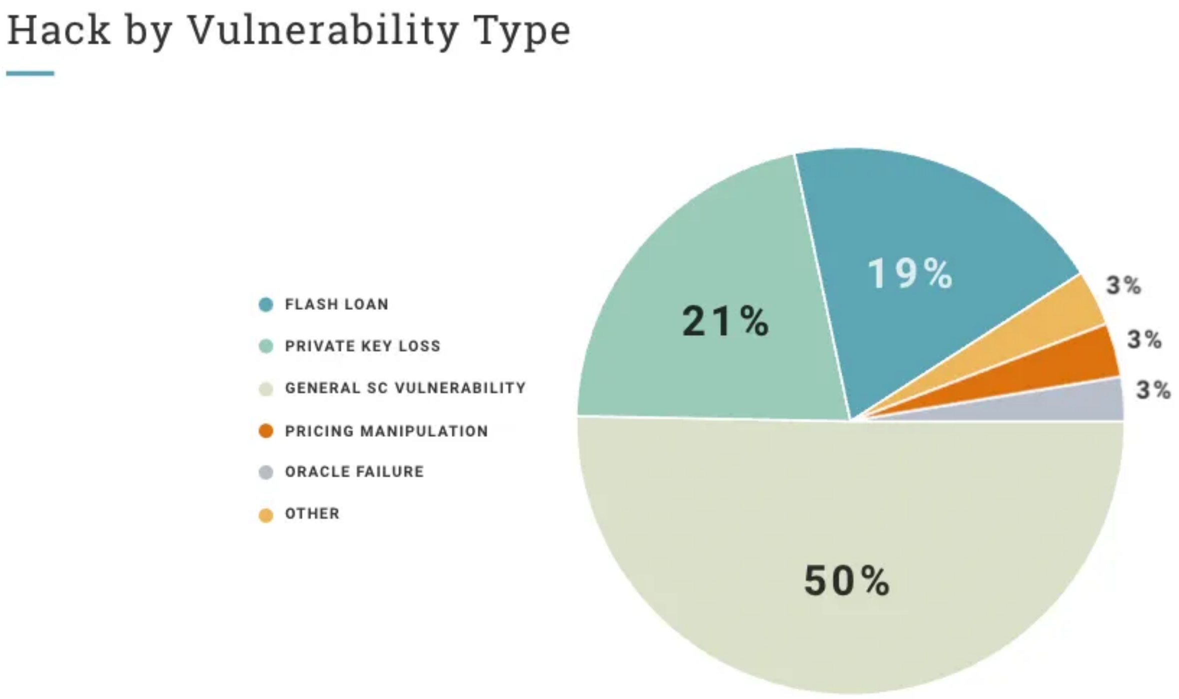
Blessings for Protocols

1. **Increased Liquidity Efficiency (for the provided Flash Loan service)**
2. **Revenue Generation**
3. **Enhanced Market Efficiency**

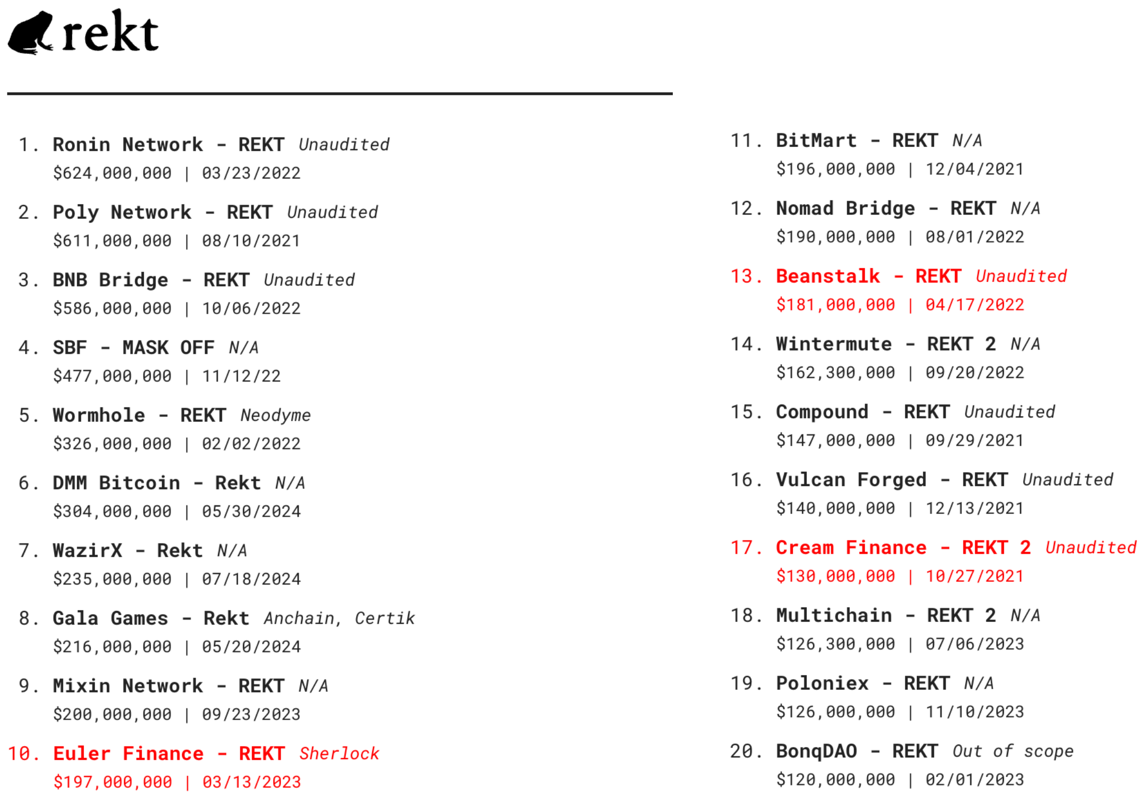
The Cures of Flash Loan

“It's like a disaster happened in just a few seconds.”

#HITB2024BKK



Ref: https://medium.com/@cicadapartners/smart-contract-vulnerabilities-in-on-chain-lending-3d6287ce9494

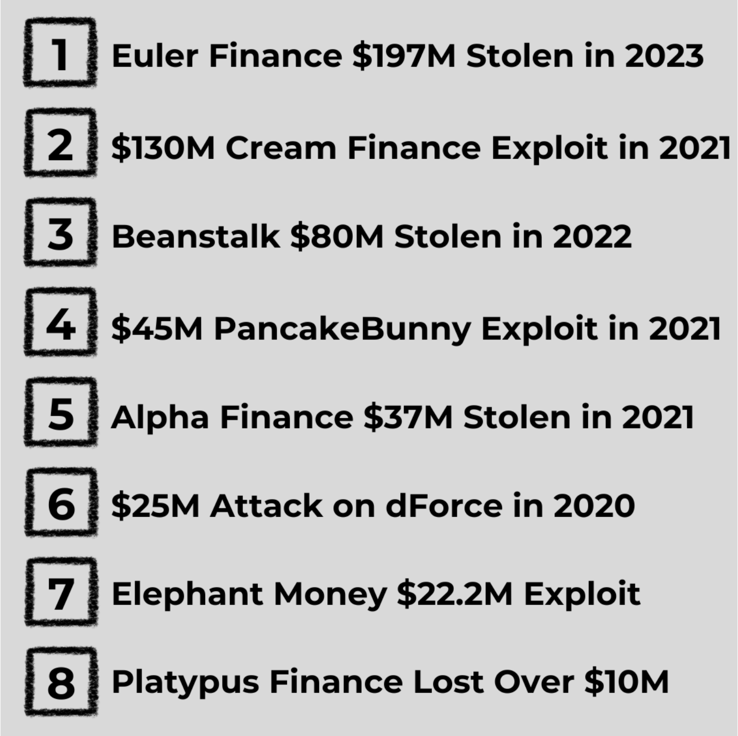


rekt

1. **Ronin Network - REKT** *Unaudited*
   $624,000,000 | 03/23/2022
2. **Poly Network - REKT** *Unaudited*
   $611,000,000 | 08/10/2021
3. **BNB Bridge - REKT** *Unaudited*
   $586,000,000 | 10/06/2022
4. **SBF - MASK OFF** *N/A*
   $477,000,000 | 11/12/22
5. **Wormhole - REKT** *Neodyme*
   $326,000,000 | 02/02/2022
6. **DMM Bitcoin - Rekt** *N/A*
   $304,000,000 | 05/30/2024
7. **WazirX - Rekt** *N/A*
   $235,000,000 | 07/18/2024
8. **Gala Games - Rekt** *Anchain, Certik*
   $216,000,000 | 05/20/2024
9. **Mixin Network - REKT** *N/A*
   $200,000,000 | 09/23/2023
10. **Euler Finance - REKT** *Sherlock*
    $197,000,000 | 03/13/2023
11. **BitMart - REKT** *N/A*
    $196,000,000 | 12/04/2021
12. **Nomad Bridge - REKT** *N/A*
    $190,000,000 | 08/01/2022
13. **Beanstalk - REKT** *Unaudited*
    $181,000,000 | 04/17/2022
14. **Wintermute - REKT 2** *N/A*
    $162,300,000 | 09/20/2022
15. **Compound - REKT** *Unaudited*
    $147,000,000 | 09/29/2021
16. **Vulcan Forged - REKT** *Unaudited*
    $140,000,000 | 12/13/2021
17. **Cream Finance - REKT 2** *Unaudited*
    $130,000,000 | 10/27/2021
18. **Multichain - REKT 2** *N/A*
    $126,300,000 | 07/06/2023
19. **Poloniex - REKT** *N/A*
    $126,000,000 | 11/10/2023
20. **BonqDAO - REKT** *Out of scope*
    $120,000,000 | 02/01/2023

Ref: https://rekt.news/leaderboard/

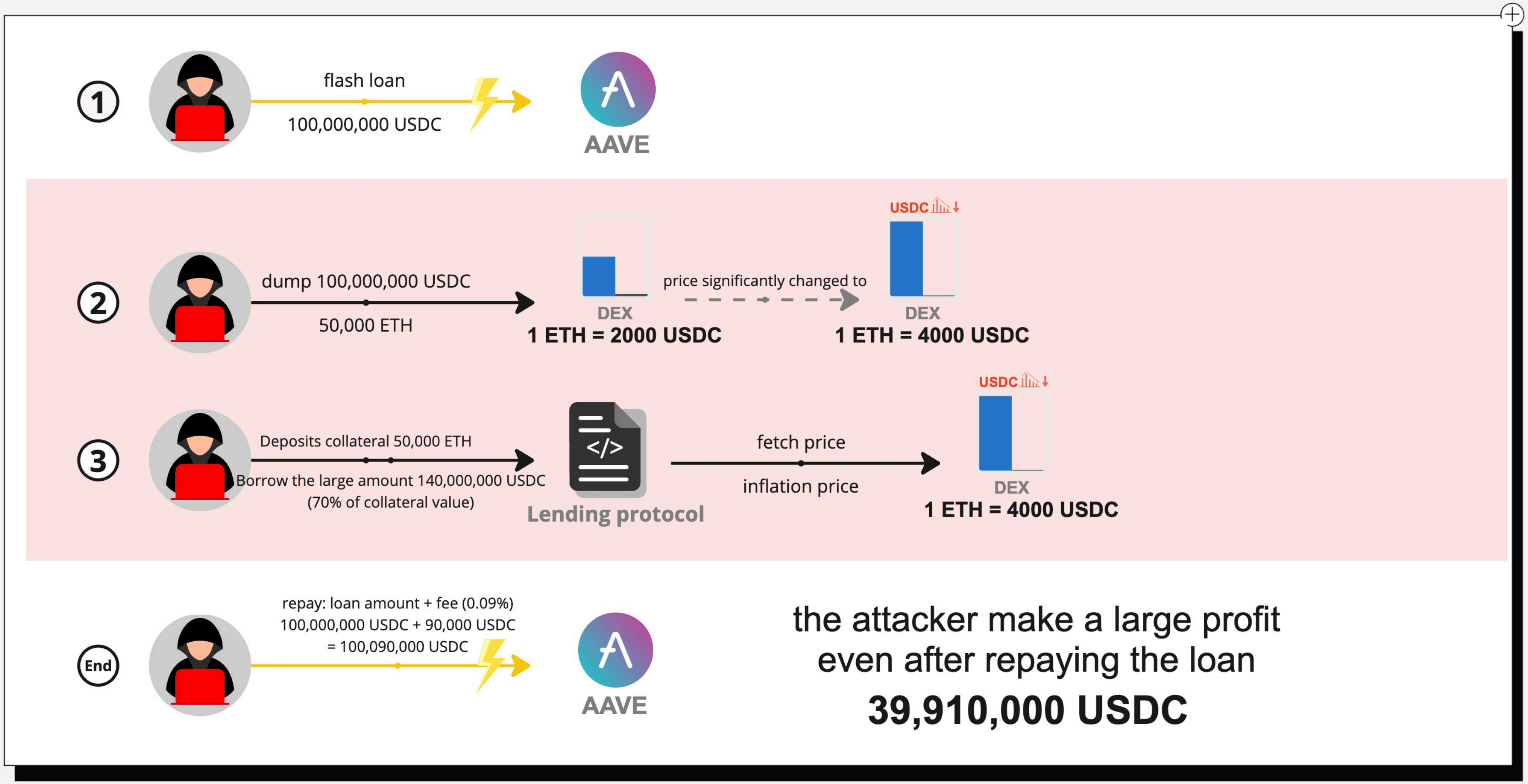
#HITB2024BKK



1. Euler Finance $197M Stolen in 2023
2. $130M Cream Finance Exploit in 2021
3. Beanstalk $80M Stolen in 2022
4. $45M PancakeBunny Exploit in 2021
5. Alpha Finance $37M Stolen in 2021
6. $25M Attack on dForce in 2020
7. Elephant Money $22.2M Exploit
8. Platypus Finance Lost Over $10M

Ref: https://bitcoin.tax/blog/biggest-crypto-flash-loan-attacks/

How to do the Flash Loan Attack



1 flash loan 100,000,000 USDC AAVE

2 dump 100,000,000 USDC 50,000 ETH DEX 1 ETH = 2000 USDC price significantly changed to USDC DEX 1 ETH = 4000 USDC

3 Deposits collateral 50,000 ETH Borrow the large amount 140,000,000 USDC (70% of collateral value) Lending protocol fetch price inflation price USDC DEX 1 ETH = 4000 USDC

End repay: loan amount + fee (0.09%) 100,000,000 USDC + 90,000 USDC = 100,090,000 USDC AAVE

the attacker make a large profit even after repaying the loan

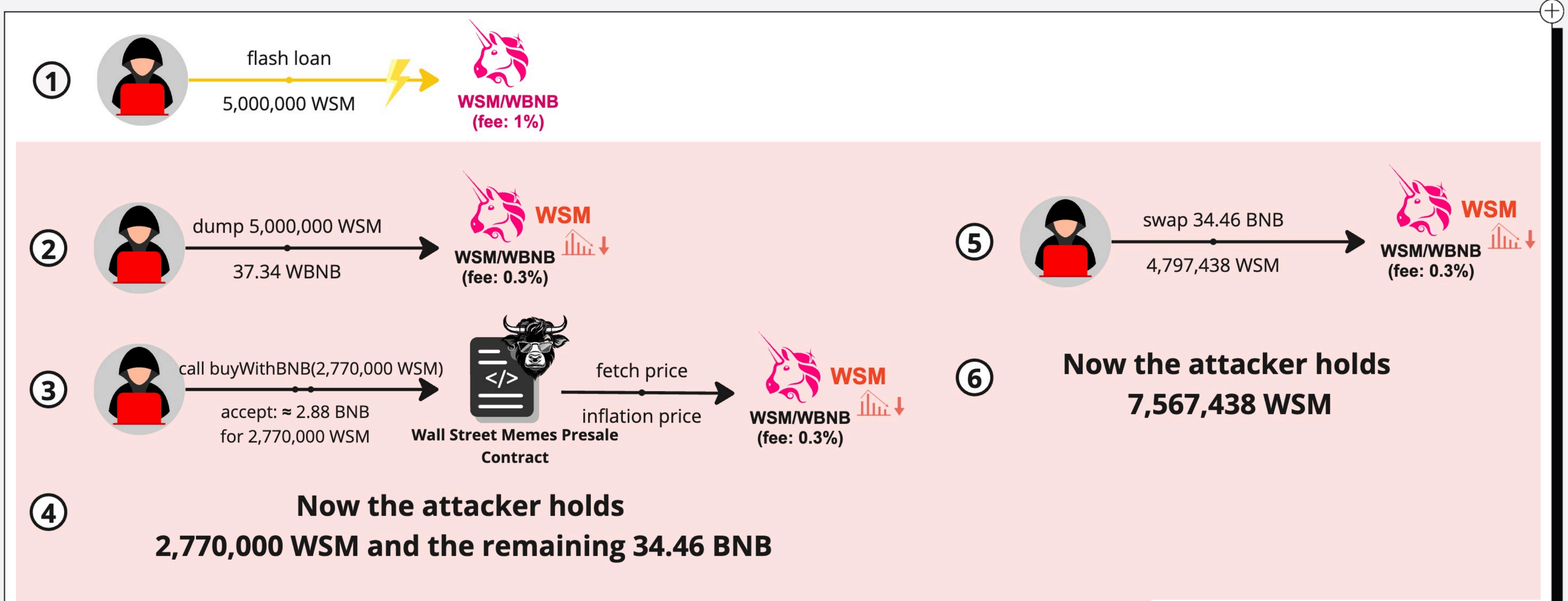
39,910,000 USDC

#HITB2024BKK

Demo#1 Zunami

April, 2024

Loss ≈ $18,000



1 flash loan 5,000,000 WSM → WSM/WBNB (fee: 1%)

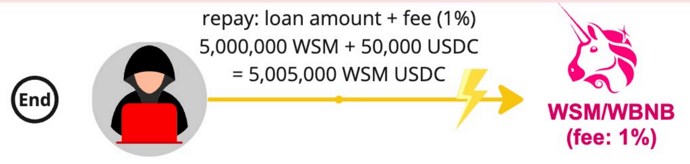
2 dump 5,000,000 WSM 37.34 WBNB → WSM WSM/WBNB (fee: 0.3%)

3 call buyWithBNB(2,770,000 WSM) accept: ≈ 2.88 BNB for 2,770,000 WSM → Wall Street Memes Presale Contract fetch price inflation price → WSM WSM/WBNB (fee: 0.3%)

4 **Now the attacker holds 2,770,000 WSM and the remaining 34.46 BNB**

5 swap 34.46 BNB 4,797,438 WSM → WSM WSM/WBNB (fee: 0.3%)

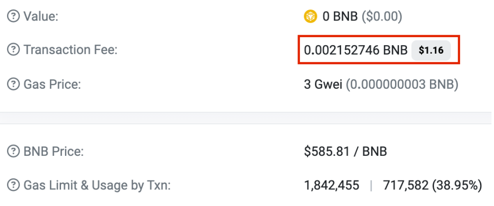
6 **Now the attacker holds 7,567,438 WSM**



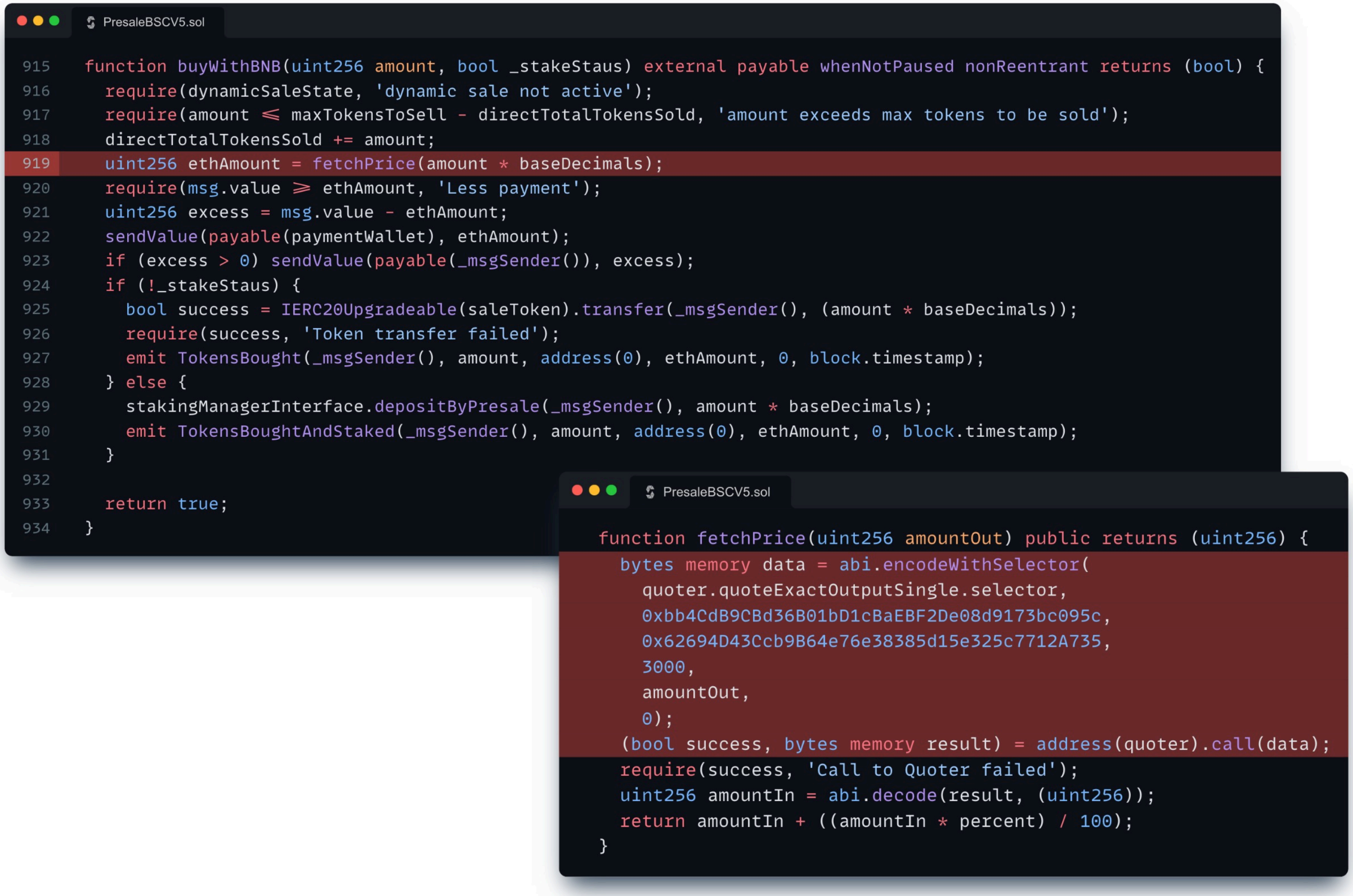
End repay: loan amount + fee (1%) 5,000,000 WSM + 50,000 USDC = 5,005,000 WSM USDC → WSM/WBNB (fee: 1%)

the attacker make a profit after repaying the loan

≈ 2,517,438 WSM (≈ $18,000)



ⓘ Value:	0 BNB ($0.00)
ⓘ Transaction Fee:	0.002152746 BNB $1.16
ⓘ Gas Price:	3 Gwei (0.000000003 BNB)
ⓘ BNB Price:	$585.81 / BNB
ⓘ Gas Limit & Usage by Txn:	1,842,455 | 717,582 (38.95%)



PresaleBSCV5.sol

```solidity
function buyWithBNB(uint256 amount, bool _stakeStaus) external payable whenNotPaused nonReentrant returns (bool) {
  require(dynamicSaleState, 'dynamic sale not active');
  require(amount <= maxTokensToSell - directTotalTokensSold, 'amount exceeds max tokens to be sold');
  directTotalTokensSold += amount;
  uint256 ethAmount = fetchPrice(amount * baseDecimals);
  require(msg.value >= ethAmount, 'Less payment');
  uint256 excess = msg.value - ethAmount;
  sendValue(payable(paymentWallet), ethAmount);
  if (excess > 0) sendValue(payable(_msgSender()), excess);
  if (!_stakeStaus) {
    bool success = IERC20Upgradeable(saleToken).transfer(_msgSender(), (amount * baseDecimals));
    require(success, 'Token transfer failed');
    emit TokensBought(_msgSender(), amount, address(0), ethAmount, 0, block.timestamp);
  } else {
    stakingManagerInterface.depositByPresale(_msgSender(), amount * baseDecimals);
    emit TokensBoughtAndStaked(_msgSender(), amount, address(0), ethAmount, 0, block.timestamp);
  }

  return true;
}
```

PresaleBSCV5.sol

```solidity
function fetchPrice(uint256 amountOut) public returns (uint256) {
  bytes memory data = abi.encodeWithSelector(
    quoter.quoteExactOutputSingle.selector,
    0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c,
    0x62694D43Ccb9B64e76e38385d15e325c7712A735,
    3000,
    amountOut,
    0);
  (bool success, bytes memory result) = address(quoter).call(data);
  require(success, 'Call to Quoter failed');
  uint256 amountIn = abi.decode(result, (uint256));
  return amountIn + ((amountIn * percent) / 100);
}
```

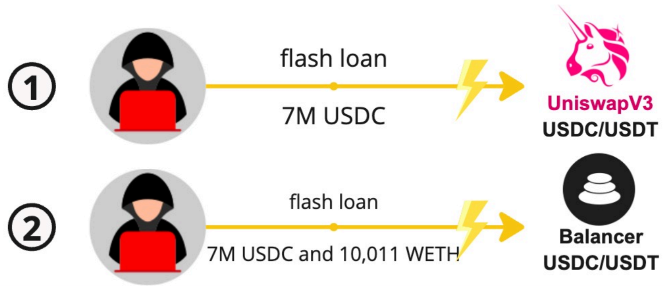
#HITB2024BKK

Demo#2 Zunami

August, 2023

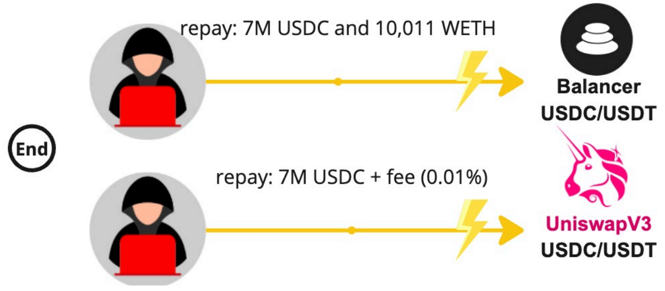
Loss ≈ $2.16 million

#HITB2024BKK



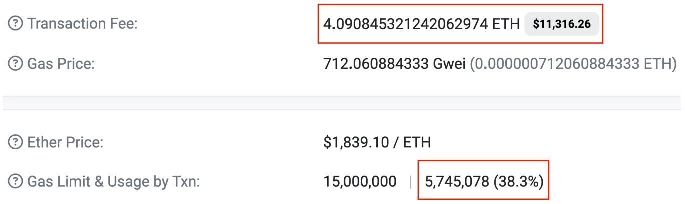
Now the attacker holds
2,770,000 USDC and 10,011 WETH

Go to the code
and
Transaction



the attacker make a profit after repaying the loan

1,152 WETH (≈ $2.12M)



ⓘ Transaction Fee:	4.090845321242062974 ETH $11,316.26
ⓘ Gas Price:	712.060884333 Gwei (0.000000712060884333 ETH)
ⓘ Ether Price:	$1,839.10 / ETH
ⓘ Gas Limit & Usage by Txn:	15,000,000 | 5,745,078 (38.3%)

How to Prevent Flash Loan Attacks

How to Prevent Flash Loan Attacks

- **Avoid using spot price (use TWAP, VWAP instead)**
- **Use Off-Chain Oracles for Price Data (Chainlink, Pyth, Band Protocol)**
- **Force Critical Transactions to Go Through Two Blocks**
- **Conduct Smart Contract Audit**
- **Conduct Economic Audit**
- **Monitoring and Alerting**
- **Incident response plan**

Q & A

ValiX
Consulting

✉ info@valix.io

www.valix.io

fb.com/valixconsulting

twitter.com/valixconsulting

medium.com/valixconsulting

#HITB2024BKK