CCR Stable Token

Challenge Task

Block-Chain 2024

Gruppe 3

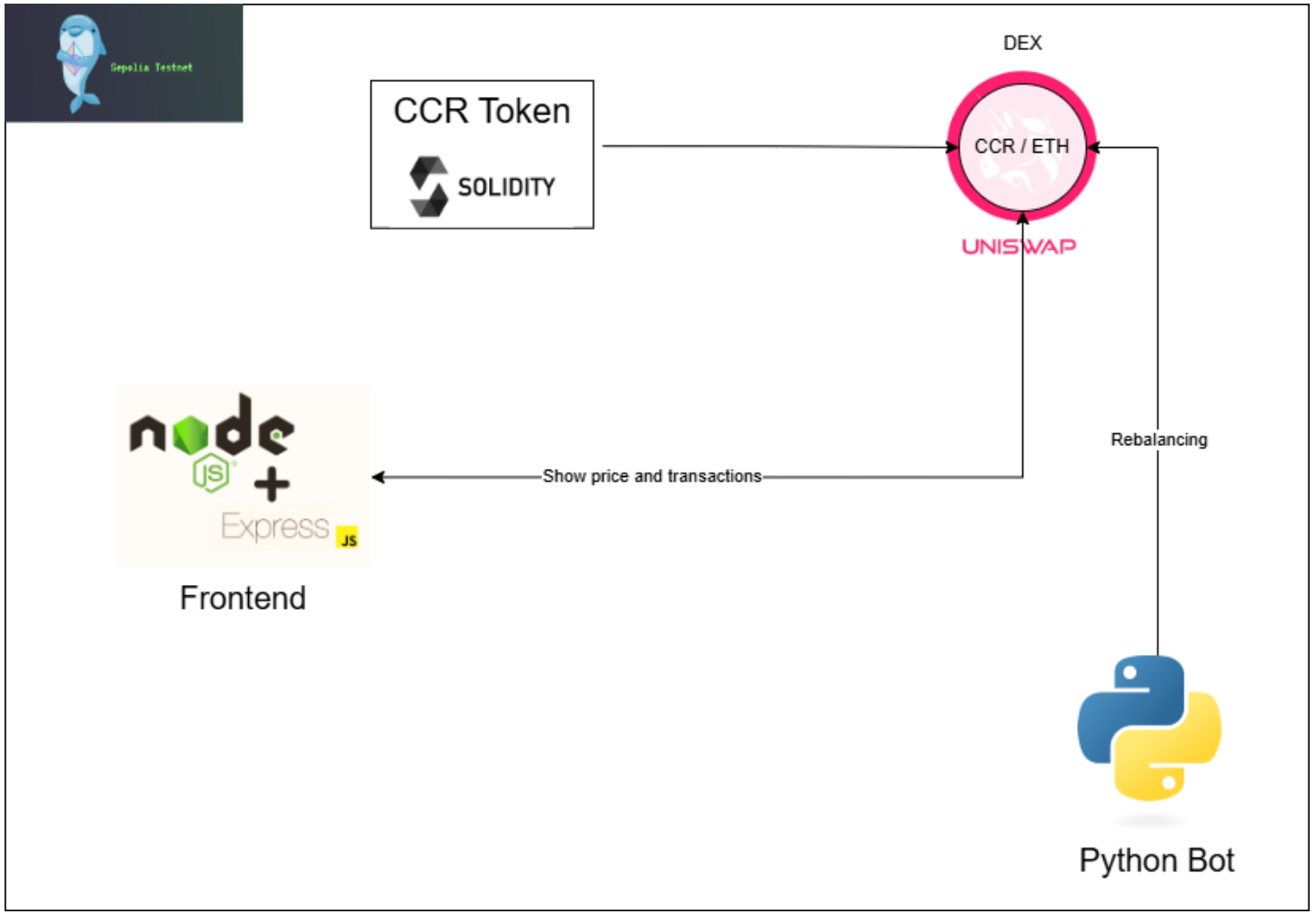
Cedric Christen, Corsin Salutt, Roman Weber

10 December 2024

OST
Eastern Switzerland
University of Applied Sciences

Inhaltsverzeichnis

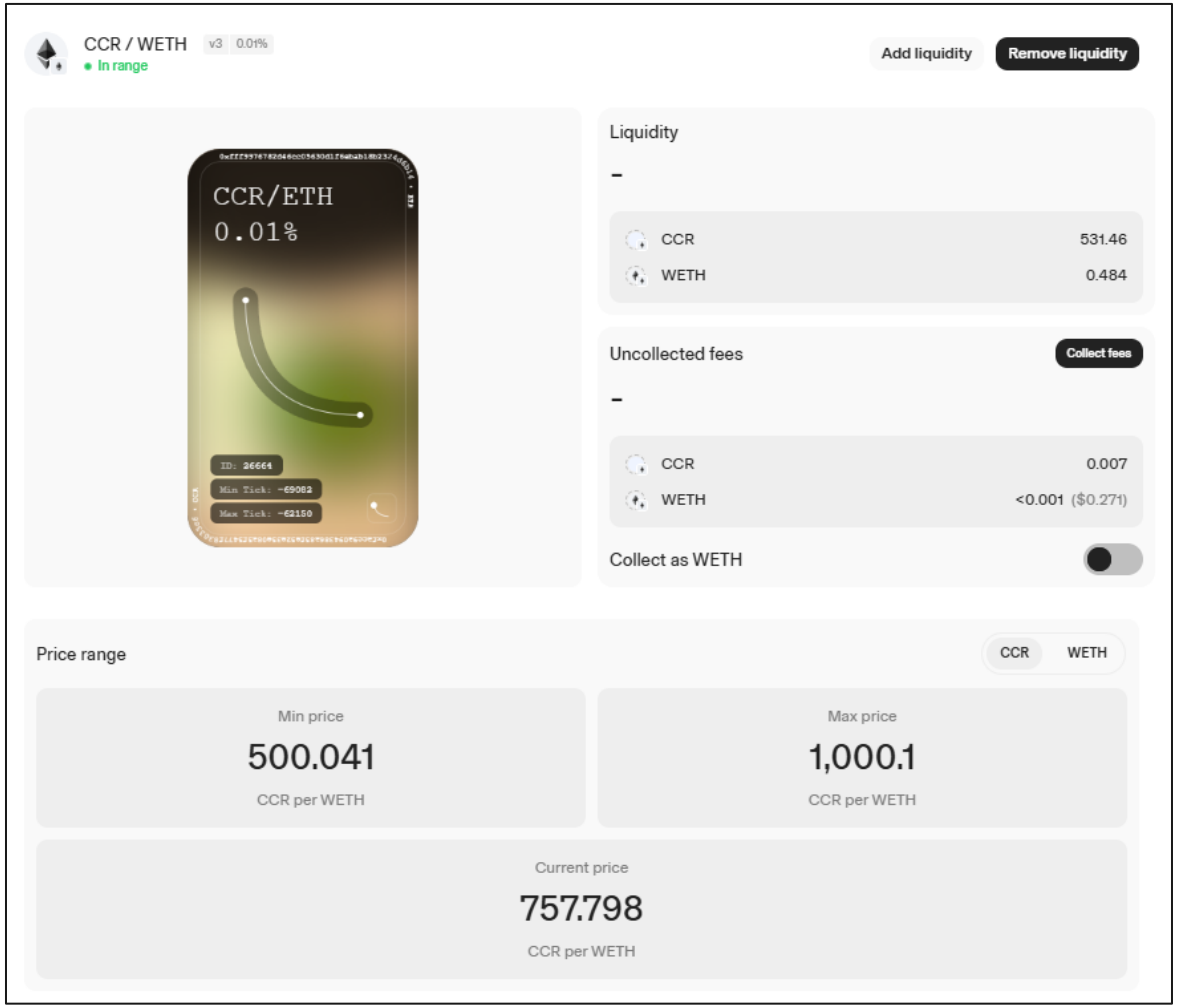
Architecture



Solidity

- Base for CCR Token
- OpenZeppelin Modules
  - ERC20 format
  - Burnable
  - Ownable

Uniswap


CCR / WETH
v3
0.01%
In range
Add liquidity
Remove liquidity
CCR/ETH
0.01%
ID: 26664
Min Tick: -69082
Max Tick: -62150
Liquidity
-
CCR
531.46
WETH
0.484
Uncollected fees
Collect fees
-
CCR
0.007
WETH
<0.001 ($0.271)
Collect as WETH
Price range
CCR
WETH
Min price
500.041
CCR per WETH
Max price
1,000.1
CCR per WETH
Current price
757.798
CCR per WETH

Python-bot

Reasons for Python bot

- Struggled with other approaches
- Seemed straight forward

Specifics

- Access to a wallet
  - Funds on wallet used for rebalancing
- Target ratio: **1 WETH is 1000 CCR** token
- Ratio Threshold: 10
  - Exact ration practically impossible to achieve

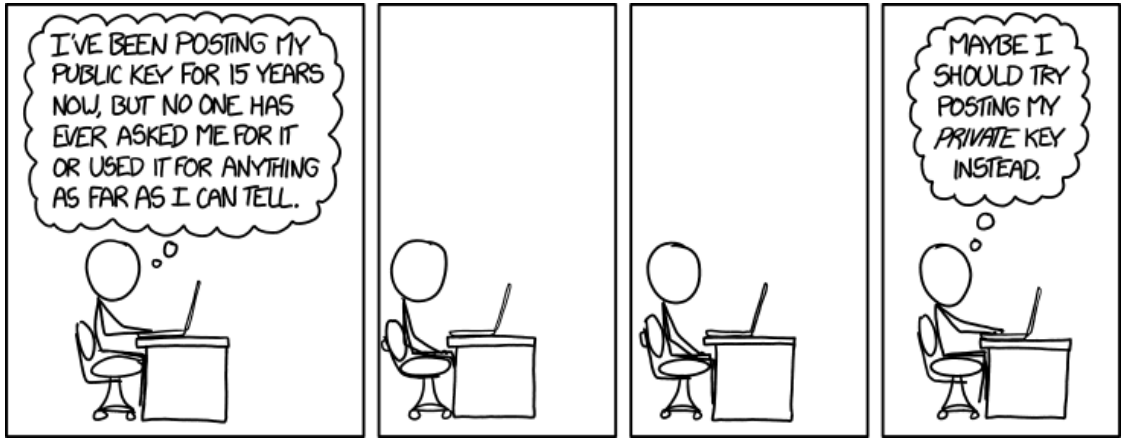
How does the bot work?

1. Loads adresses of pool, tokens, router, ABIs and contracts.
2. Balances of CCR, WETH and ratio calculated
3. If ratio is +-10, tokens added to pool
4. Calculates CCR or WETH amount to add to the pool.
5. Transaction approval for Uniswap router
6. ExactInputSingle with the calculated value.
7. Waits for one minute, then starts at 2. again.

Python-bot

Future improvements

- Don't commit private key!
- Optimize to_add_amount algorithm
  - To_add_amount * 0.7
  - Over- and undershoots multiple times
  - Each transaction has costs
  - Algorithm works well enough for testat



https://xkcd.com/1553/

Frontend

- Written in javascript using Node.js
- Main Dependencies:
  - Express with Handlebars
  - Web3 for the connection with the Contracts

Frontend

- Requirements for the Connection to the Contracts.
  - Gateway address of the Router
  - Contract Address
  - Json ABI

Frontend

- Getting the price from the pool:
  - Slot0 and then the sqrtPricex96
    - This price is extended to 96 bits
    - Transform the price with: (sqrtPriceX96 ** 2) / (2 ** 192)

- Getting the Token history
  - GetPastEvents returns a list of past events
    - Use filter Transaction to only get the transactions
    - Format it and present it

Demo

- Frontend
- Starting python-bot
- Adding to pool with transfer
- Show rebalancing
- Changes in Frontend