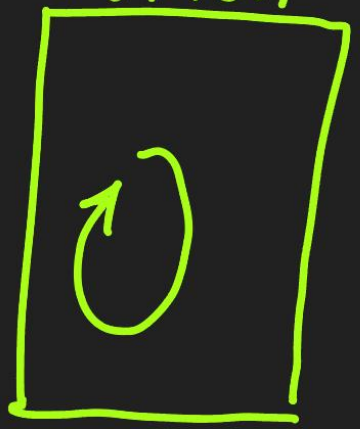
```
/**
 * @dev Helper to calculate interest.
 * @param principal Amount in STETH.
 * @param timeElapsed Time since deposit (in seconds).
 * @param annualRate Annual interest  in bps.
 */
function calculateInterest(
    uint256 principal,
    uint256 timeElapsed,
    uint256 annualRate
) public pure returns (  56
    uint256 timeInYears = timeElapsed * 1e18 / 365 days; // Convert t
    uint256 rate = ann                                  0; // Convert bps to fixe
    return (principal * rate * timeInYears) / 1e18; // Fixed-point ir
}

/*
 * @dev Set new oracle address (Owner only).
 */
function setOracle(address _oracle) external onlyOw
    oracle = IOracle(_oracle);
}
```

Python



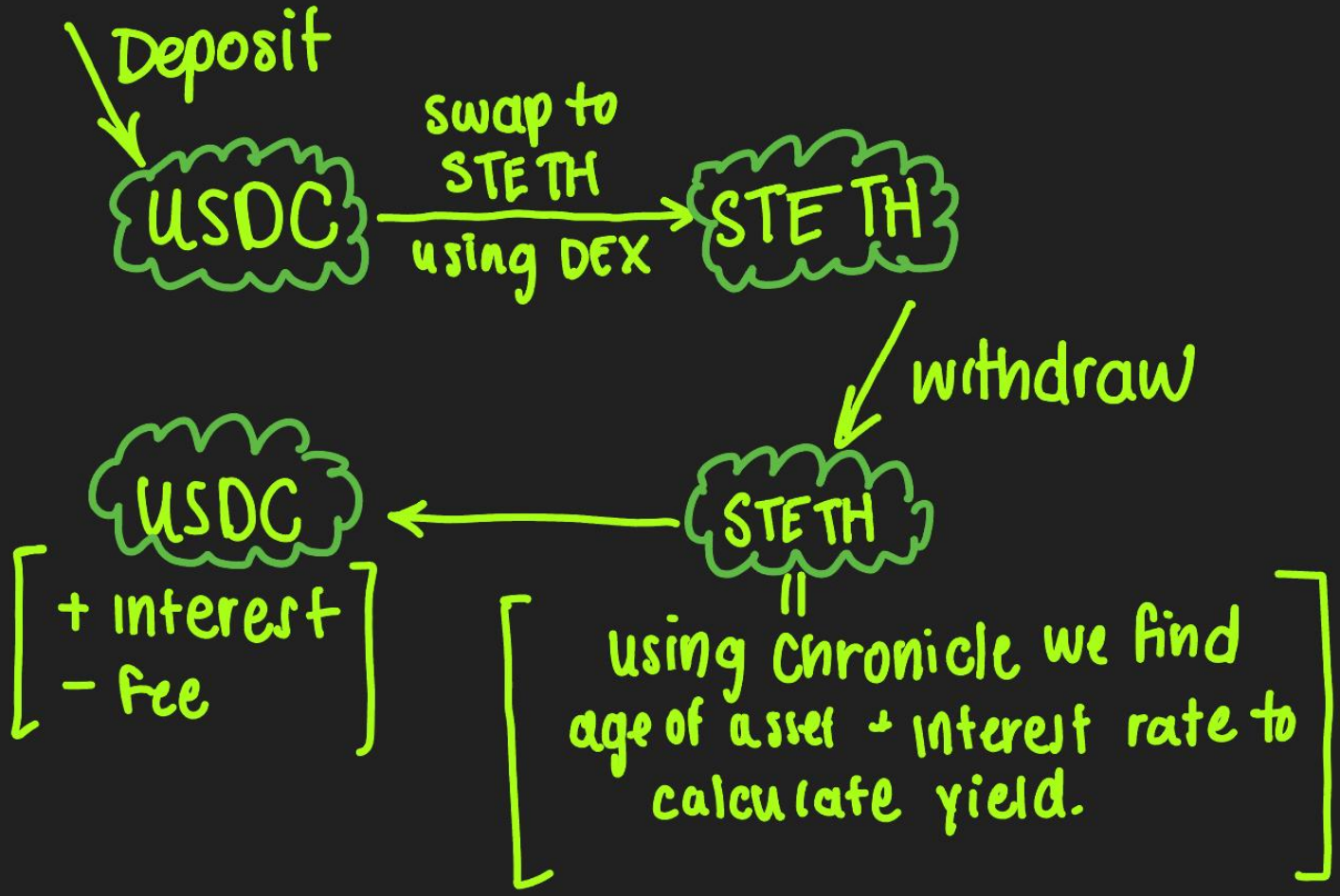
- timed loop to check interest.

* maybe add logic to trade between ETH + USDC (?)

SOLIDITY

- deposit USDC, convert to STETH (ETH + interest).

- withdraw converts back to USDC (& returns interest).


Deposit
USDC
swap to STETH using DEX
STETH
withdraw
USDC
+ interest
- fee
STETH
using Chronicle we find age of asset + interest rate to calculate yield.

Future Ideas:

* Leverage highs + lows of USDC and ETH to develope simple strategy to trade between USDC and STETH rather than soley relying on interest.

```
/**
 * @dev Helper to calculate interest.
 * @param principal Amount in STETH.
 * @param timeElapsed Time since deposit (in seconds).
 * @param annualRate Annual interest  in bps.
 */
function calculateInterest(
    uint256 principal,
    uint256 timeElapsed,
    uint256 annualRate
) public pure returns (  56
    uint256 timeInYears = timeElapsed * 1e18 / 365 days; // Convert t
    uint256 rate = ann                                  0; // Convert bps to fixe
    return (principal * rate * timeInYears) / 1e18; // Fixed-point ir
}

/*
 * @dev Set new oracle address (Owner only).
 */
function setOracle(address _oracle) external onlyOw
    oracle = IOracle(_oracle);
}
```