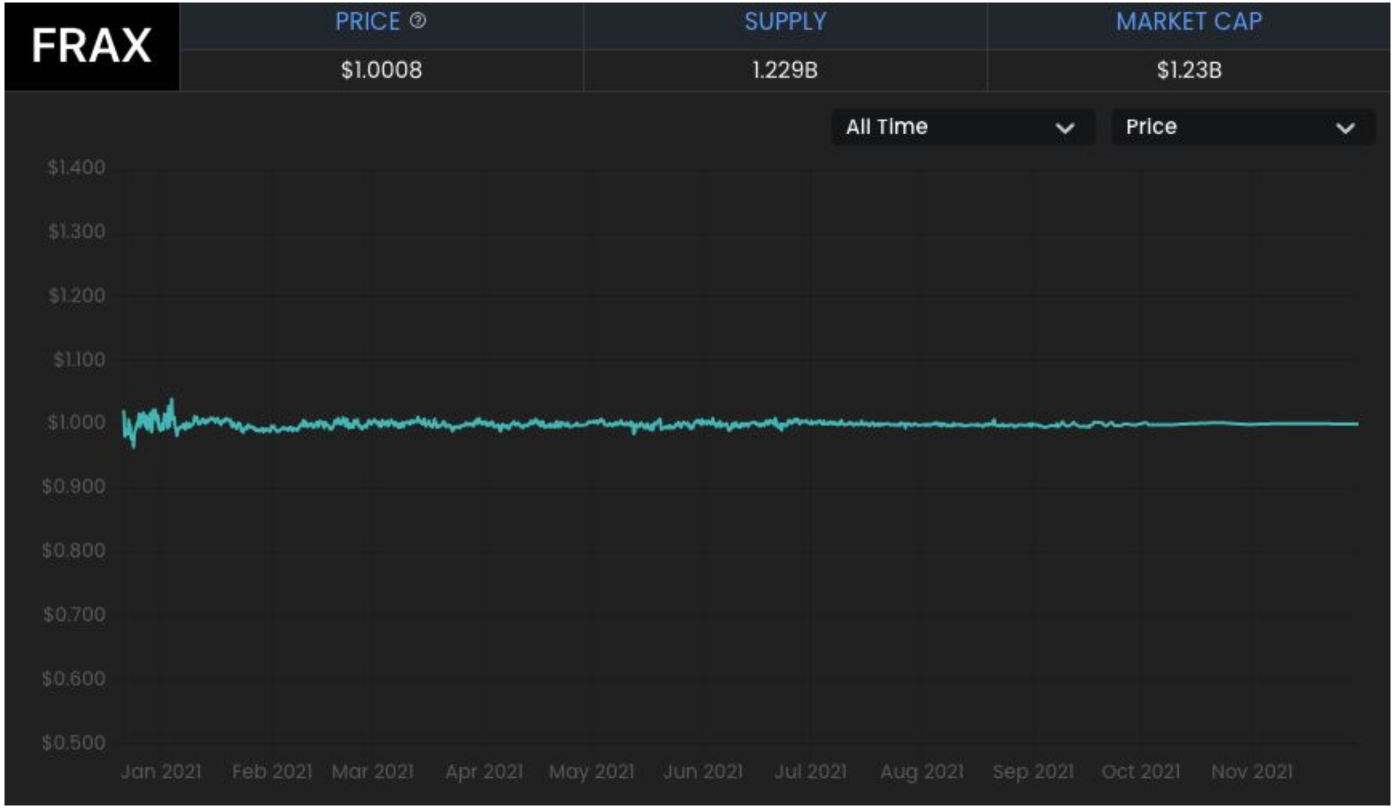
Olympus

Olympus is a decentralized reserve currency protocol
Each OHM token is backed by a basket of assets (e.g. DAI, FRAX) in the Olympus treasury

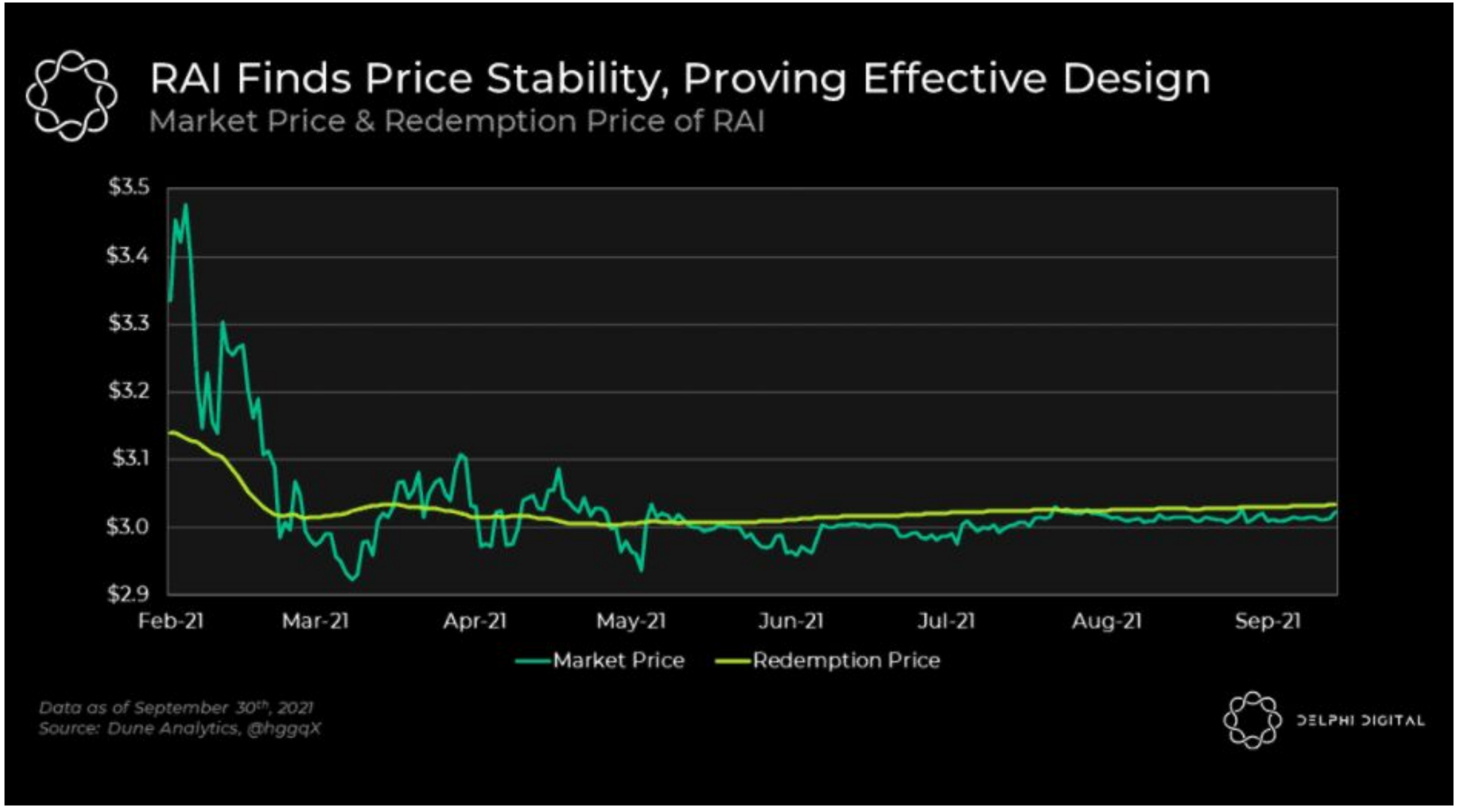
Algorithmic Stablecoins

1. ESD/DSD - Bond
2. Basis/MITH - Two Protocol tokens
3. Frax/UST - 1:1 peg
4. Fei - PCV
5. Rai - Control theory

Frax Peg


FRAX
PRICE
SUPPLY
MARKET CAP
$1.0008
1.229B
$1.23B
All Time
Price
$1.400
$1.300
$1.200
$1.100
$1.000
$0.900
$0.800
$0.700
$0.600
$0.500
Jan 2021
Feb 2021
Mar 2021
Apr 2021
May 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
Nov 2021

Rai Stability



So what’s next?

1. Perhaps with regulators turning their attention to dollar-pegged stablecoins, the utility and vision of successful algo stablecoins will start to resonate more with the DeFi community - highly resilient, decentralized
2. Utility value?
   a. DAI/MIM/LUSD - lending/leverage
   b. USDT/USDC - spot trading pair/perp/collateral/ramp

OHM

Borrow a lot from the designs of algo stablecoins

1. Bond
2. PCV

Break a critical rule: 1 OHM is backed but not pegged at $1 which means:

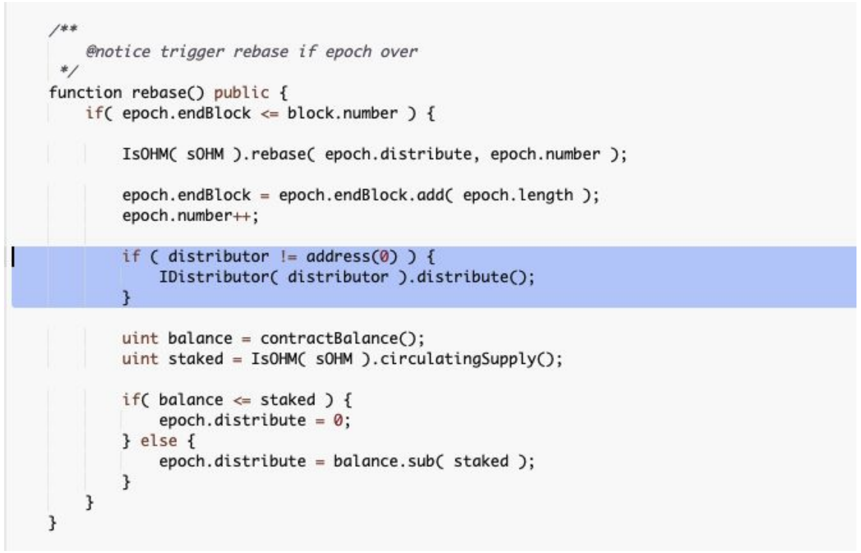
1. Expansions can be controlled at a reasonable scale (Controlled APY)
2. Do not have target price - decline exception when the price > 1

OHM Explained

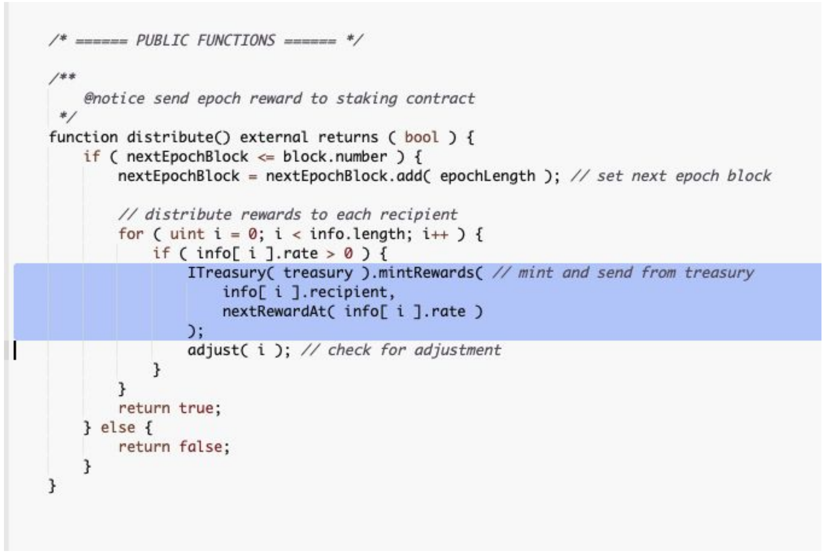
1. Data
2. Workflow
3. Key features
   a. Staking
   b. Bonding

Staking - High APY

Rebase: OlympusStaking, Distributor



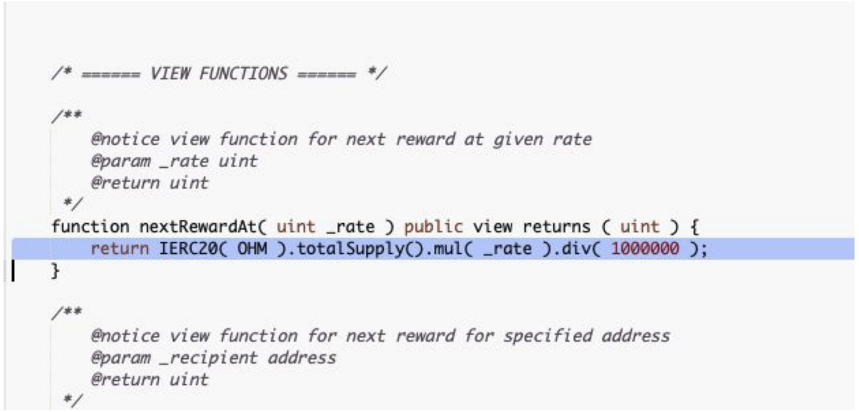
```
/**
    @notice trigger rebase if epoch over
 */
function rebase() public {
    if( epoch.endBlock <= block.number ) {

        IsOHM( sOHM ).rebase( epoch.distribute, epoch.number );

        epoch.endBlock = epoch.endBlock.add( epoch.length );
        epoch.number++;

        if ( distributor != address(0) ) {
            IDistributor( distributor ).distribute();
        }

        uint balance = contractBalance();
        uint staked = IsOHM( sOHM ).circulatingSupply();

        if( balance <= staked ) {
            epoch.distribute = 0;
        } else {
            epoch.distribute = balance.sub( staked );
        }
    }
}
```



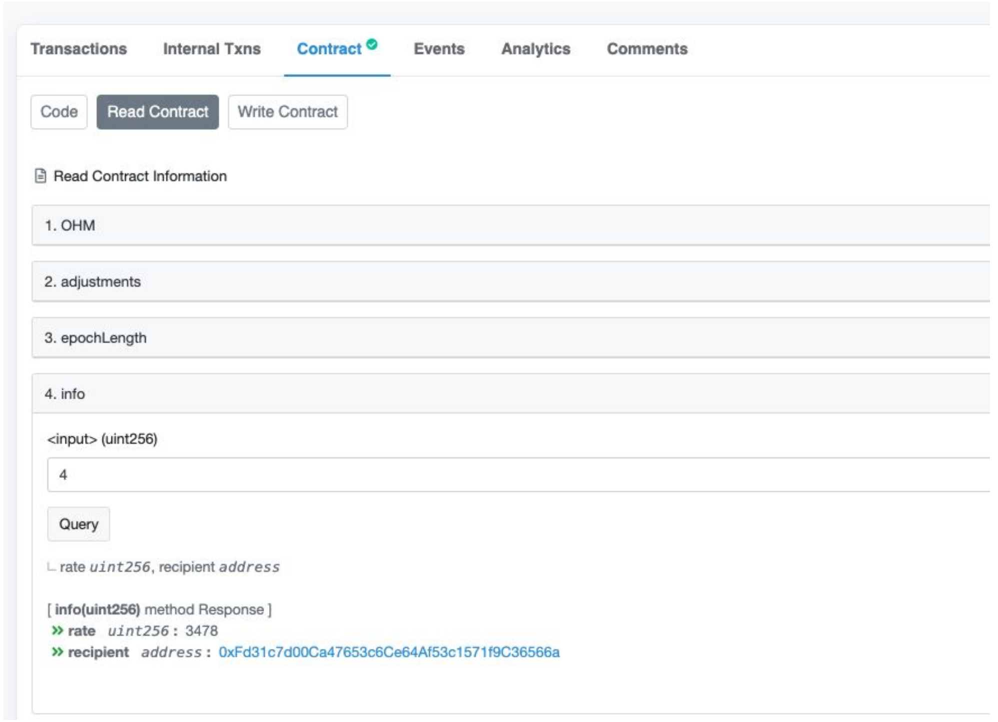
```
/* ====== PUBLIC FUNCTIONS ====== */

/**
    @notice send epoch reward to staking contract
 */
function distribute() external returns ( bool ) {
    if ( nextEpochBlock <= block.number ) {
        nextEpochBlock = nextEpochBlock.add( epochLength ); // set next epoch block

        // distribute rewards to each recipient
        for ( uint i = 0; i < info.length; i++ ) {
            if ( info[ i ].rate > 0 ) {
                ITreasury( treasury ).mintRewards( // mint and send from treasury
                    info[ i ].recipient,
                    nextRewardAt( info[ i ].rate )
                );
                adjust( i ); // check for adjustment
            }
        }
        return true;
    } else {
        return false;
    }
}
```

Staking - APY - x% inflation of total supply per 8 hours

Rebase:[Distributor]



```
/* ====== VIEW FUNCTIONS ====== */

/**
    @notice view function for next reward at given rate
    @param _rate uint
    @return uint
 */
function nextRewardAt( uint _rate ) public view returns ( uint ) {
    return IERC20( OHM ).totalSupply().mul( _rate ).div( 1000000 );
}

/**
    @notice view function for next reward for specified address
    @param _recipient address
    @return uint
 */
```



Read Contract Information

1. OHM

2. adjustments

3. epochLength

4. info

<input> (uint256)

4

∟ rate *uint256*, recipient *address*

[**info(uint256)** method Response]
» **rate** *uint256* : 3478
» **recipient** *address* : 0xFd31c7d00Ca47653c6Ce64Af53c1571f9C36566a

Staking - APY Calc

1. x% * Total supply / Staking OHM amount = apr in every 8 hrs
2. How to determine x%? Algo?
3. Why choose this way? Highly controlled numbers - not too high nor too low

Bonding

1. Buying by liquidity - PCV
   a. Self controlled liquidity vs Rented liquidity (death spiral) - defi 2.0 (how does protocol acquire liquidity (pool2))
   b. Increased reserve fund to back OHM
2. High acquisition cost - users need to buy at a discounted price instead of liquidity mining

Olympus Pro

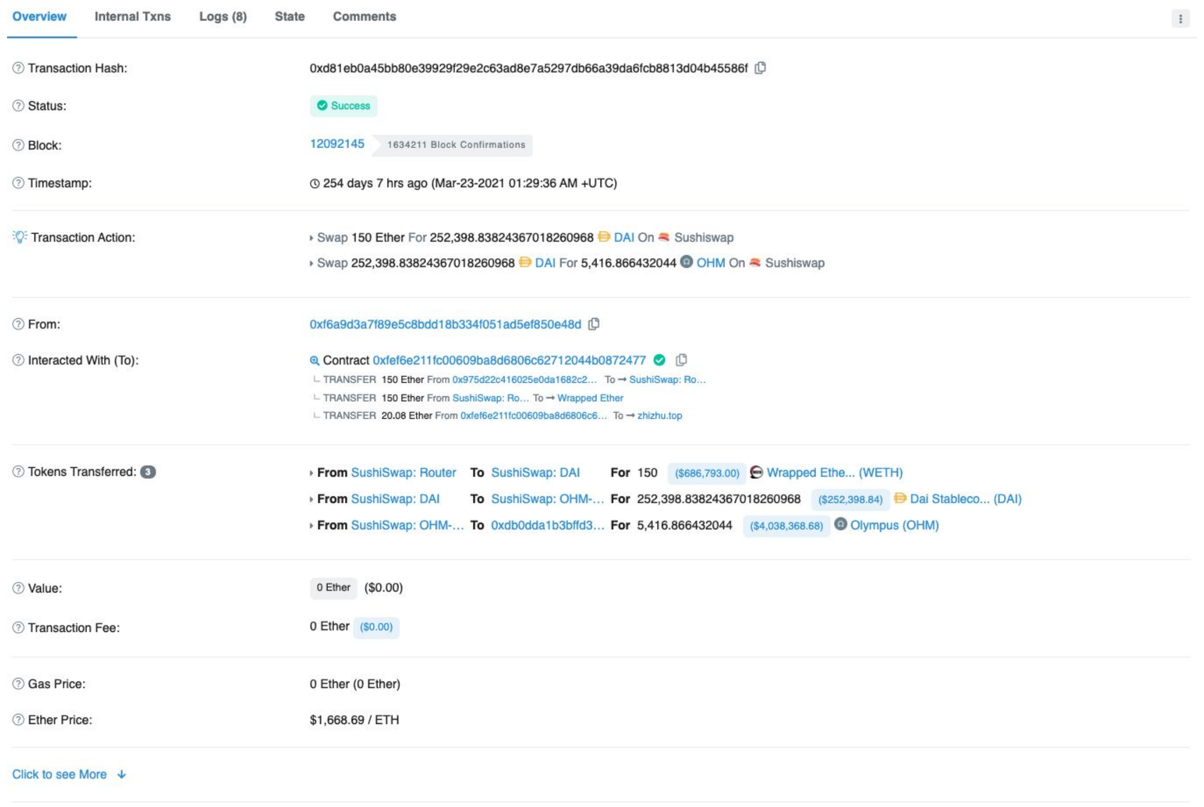
Bonding as a service

OHM IDO - initial discord offering

1. No VC, NO liquidity mining, No airdrop
2. 50,000 OHM at the beginning
    a. 73% IDO for discord members, price: $4 for every members - no whales
    b. 27% for SLP init LP

40 * 200 = 8000x

OHM IDO - initial discord offering at 3/23 - tx



Overview | Internal Txns | Logs (8) | State | Comments

Transaction Hash:	0xd81eb0a45bb80e39929f29e2c63ad8e7a5297db66a39da6fcb8813d04b45586f
Status:	Success
Block:	12092145 1634211 Block Confirmations
Timestamp:	254 days 7 hrs ago (Mar-23-2021 01:29:36 AM +UTC)
Transaction Action:	Swap 150 Ether For 252,398.83824367018260968 DAI On Sushiswap Swap 252,398.83824367018260968 DAI For 5,416.866432044 OHM On Sushiswap
From:	0xf6a9d3a7f89e5c8bdd18b334f051ad5ef850e48d
Interacted With (To):	Contract 0xfef6e211fc00609ba8d6806c62712044b0872477 ✓ └ TRANSFER 150 Ether From 0x975d22c416025e0da1682c2… To → SushiSwap: Ro… └ TRANSFER 150 Ether From SushiSwap: Ro… To → Wrapped Ether └ TRANSFER 20.08 Ether From 0xfef6e211fc00609ba8d6806c6… To → zhizhu.top
Tokens Transferred: 3	From SushiSwap: Router To SushiSwap: DAI For 150 ($686,793.00) Wrapped Ethe... (WETH) From SushiSwap: DAI To SushiSwap: OHM-… For 252,398.83824367018260968 ($252,398.84) Dai Stableco... (DAI) From SushiSwap: OHM-… To 0xdb0dda1b3bffd3… For 5,416.866432044 ($4,038,368.68) Olympus (OHM)
Value:	0 Ether ($0.00)
Transaction Fee:	0 Ether ($0.00)
Gas Price:	0 Ether (0 Ether)
Ether Price:	$1,668.69 / ETH

Click to see More ↓