Contract interaction

Deployed contracts

Name Contract Address Staking v2 st1inch 0x9A0C8Ff858d273f57072D714bca7411D717501D7 Staking farm StakingFarmingPod 0x1A87c0F9CCA2f0926A155640e8958a8A6B0260bE Settlement Settlement 0xA88800CD213dA5Ae406ce248380802BD53b47647 Delegation pod PowerPod 0xAccfAc2339e16DC80c50d2fa81b5c2B049B4f947 Resolver information ResolverMetadata 0xBF4543819ECede56220bcB1e8C1BBa9Ef290a58a Whitelist WhitelistRegistry 0xF55684BC536487394B423e70567413faB8e45E26 Whitelist info WhitelistHelper 0xF779bdde38C39138Dcaf1514B8a6b8a6C165642D

Stake 1inch

Repository limit-order-settlement Contract st1inch.sol Contract address 0x9A0C8Ff858d273f57072D714bca7411D717501D7 Methods Staking • deposit(uint256 amount, uint256 duration) • depositWithPermit(uint256 amount, uint256 duration, bytes calldata permit) • depositFor(address account, uint256 amount, uint256 duration) • depositForWithPermit(address account, uint256 amount, uint256 duration, bytes calldata permit) | | Description | Stakes 1inch to get staking power according to the lock time |

*//Deposits 100 1inch with 1 day lock*await st1inch . deposit (ether ('100') , time . duration . days ('1')) ;

2. Register as a Resolver

2.1 Register delegation pod

Repository limit-order-settlement Contract PowerPod.sol Contract address 0x9A0C8Ff858d273f57072D714bca7411D717501D7 Methods addPod(address pod) Description Register pod usage for the tx sender. Needed for • Resolvers to enable resolver's and delegated staking power usage for whitelisting • Stakers to enable unicorn power delegation |

* // register delegation pod usage a resolver* await st1inch . connect (resolver) . addPod (delegation . address) ;

2.2. Register delegation share token

Repository limit-order-settlement Contract PowerPod.sol Contract address 0xAccfAc2339e16DC80c50d2fa81b5c2B049B4f947 Methods register(string memory name, string memory symbol, uint256 maxUserFarms) Description Creates a resolvers share token to count delegated staked power shares and accrue rewards * // Register resolver's token with name 'resolver token share' and symbol 'RTS'* await delegation . register ('resolver token share' ,

'RTS') ;

2.4. Delegate resolver's staking power to self

Repository limit-order-settlement Contract PowerPod.sol Contract address 0xDAf782667d98d5069eE7ba139932945C4D08fDE9 Methods delegate(address delegatee) Description delegate(address delegatee) * // Delegates all staking power to self* await delegation . connect (resolver) . delegate (resolver . address) ;

2.5. Whitelist resolver

Repository limit-order-settlement Contract WhitelistRegistry.sol Contract address 0xcb8308fcB7BC2f84ed1bEa2C016991D34de5cc77 Methods register() Description Checks if sender is eligible to be whitelisted and put it into the whitelist sorted by staking power descending * // Try to put the sender to the whitelist* await whitelist . connect (resolver) . register ( ) ;

3. FeeBank

Repository limit-order-settlement Contract FeeBank.sol Contract address 0xa0844e046a5B7Db55Bb8DcdFfbF0bBF9c6dc6546 Methods • deposit(uint256 amount) • depositFor(address account, uint256 amount) • depositWithPermit(uint256 amount, bytes calldata permit) • depositForWithPermit | | Description | Deposits 1inch for fee deduction when filling orders |

* // Deposit fees to fee bank* await feeBank . connect (resolver) . deposit (amount)

4. Resolving

struct

Order

{ uint256 salt ; address makerAsset ; address takerAsset ; address maker ; address receiver ; address allowedSender ;

// equals to Zero address on public orders uint256 makingAmount ; uint256 takingAmount ; uint256 offsets ; // bytes makerAssetData; // bytes takerAssetData; // bytes getMakingAmount; // this.staticcall(abi.encodePacked(bytes, swapTakerAmount)) => (swapMakerAmount) // bytes getTakingAmount; // this.staticcall(abi.encodePacked(bytes, swapMakerAmount)) => (swapTakerAmount) // bytes predicate; // this.staticcall(bytes) => (bool) // bytes permit; // On first fill: permit.1.call(abi.encodePacked(permit.selector, permit.2)) // bytes preInteraction; // bytes postInteraction; bytes interactions ;

// concat(makerAssetData, takerAssetData, getMakingAmount, getTakingAmount, predicate, permit, preIntercation, postInteraction) } Repository limit-order-settlement Contract Settlement.sol Contract address 0xA88800CD213dA5Ae406ce248380802BD53b47647 Methods .settleOrders(bytes calldata data) Description Settles an order Edit this page Previous Resolver’s setup script example Next SDK Overview