
DeFi4NFT

Release 0.0.1

Defi4 team

Jul 11, 2022

CONTENTS

1	Guide	1
1.1	Get started with DeFi4NFT	1
1.2	Neos interaction	5
1.3	Technical review	6
1.4	Install or build project	7
1.5	TroubleShooting	8
1.6	Additional features	8
1.7	Future	8
1.8	Contact	9

DeFi4NFT is a borrow/lend platform for NFTs on Algorand. At the core a state machine stores the loan health of a pure non-custodial borrow/lend protocol.

Pure non-custodial means assets remain in the owner's wallet, for borrowers this means collateral assets remain *frozen in address*. And for liquidity providers this means creating a *delegated logicSig* promising to provide a loan when a borrower matches, aka promise. And for liquidators this means a three way transaction, where the end receiver is a third party buyer willing to buy the borrower's collateral NFT and the liquidator pays the debt of the borrower while a third party buyer sends payment of the collateral NFT to liquidator. This system is set up so permissioned tokens can be open to receive liquidity from non-whitelisted addresses.

1.1 Get started with DeFi4NFT

Watch this demo video. [Go to live demo site.](#)



First step is to optin to the smartcontract.

1.1.1 Optin to DeFi4NFT

As an NFT creator

Transfer your NFT's admin address to DeFi4NFT. This will make your NFT leverageable for taking loan in DeFi4NFT dapp.

- This sets manager and freeze admin address to DeFi4NFT smartcontract
- And sets clawback to liquidator smartcontract

As a liquidity provider

Optin to the I-O-U token of DeFi4NFT dapp **dUSD**, that has 1:1 value with USDCa. This happens automatically when you create a promise to provide liquidity, via the frontend.

1.1.2 Earn (Providing Liquidity)

Choose which NFTs can borrow from your account.

- Set maximum amount you are willing to lend.
- Set expiration date for agreement.

The screenshot displays the 'Earn' section of the DeFi4NFT interface. At the top, there are four tabs: 'Borrow', 'Repay', 'Earn' (which is selected and underlined), and 'Claim'. Below the tabs, the current balance is shown as '81.200000 USDC' next to a USDC icon. There are three input fields for configuring the lending agreement:


- The first field is labeled 'Allowed NFTs' and contains the value '77141623' with a close icon (x).
- The second field is labeled 'MAX | USDC' and contains the value '0' with a close icon (x).
- The third field is labeled 'Expire Day' and contains the value '10 days'.

1.1.3 Borrow (Leveraging NFT)

Use your NFT as collateral, to borrow USDCa stablecoin.

- Set which NFT you want to collateralize
- Set amount of collateral
- Request loan You'll get requested loan amount in USDCa and your NFT will be locked.

[Borrow](#) [Repay](#) [Earn](#) [Claim](#)

 Lofty jina property 4.0 LFT-jina


MAX | | v 0 ×

MAX | USDC 0

1.1.4 Repaying loan

Send USDCa to DeFi4NFT contract. Your loan amount state will be decremented by sent repaid amount.

[Borrow](#) [Repay](#) [Earn](#) [Claim](#)

 USDC 81.200000 USDC

MAX | USDC 0 ×

| v [Amount to Repay](#) | 0

If you pay the full loan amount, your collateral assets will be unfrozen.

1.1.5 Claming USDCa

Send dUSD(I-O-U token of DeFi4NFT contract) to DeFi4NFT contract. You'll receive a 1:1 USDCa for the dUSD you send.

The screenshot shows a user interface with four tabs: Borrow, Repay, Earn, and Claim. The 'Claim' tab is active and underlined. Below the tabs, there is a balance display for 'JUSD' showing '1.130000 jUSD'. Below that is a rounded input field containing 'MAX | JUSD 0' and a close button 'X'. At the bottom center is a button labeled 'Claim USDC'.

1.1.6 Circle (Fiat-on ramp)

A fast and easy way to get fiat into our Dapp is using Circle accounts. You can either use a credit/debit card or send USDC from other chains supported by circle.

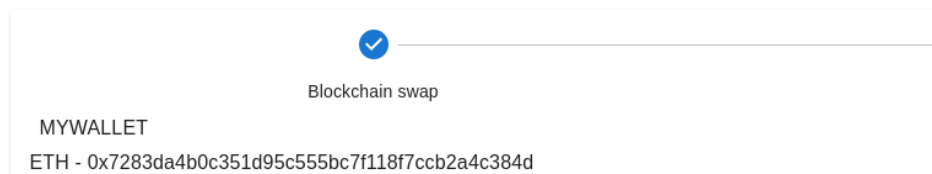
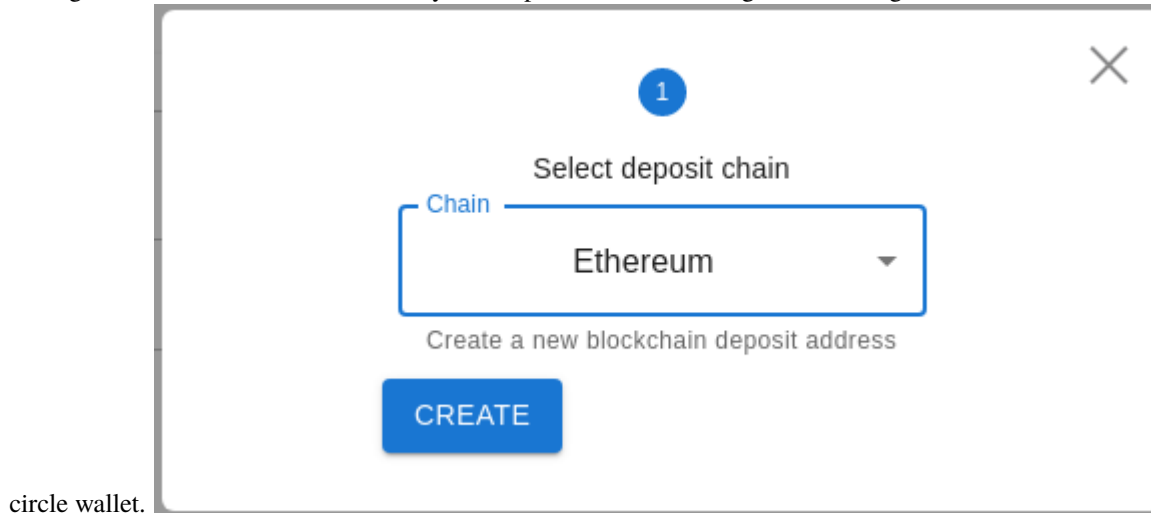
Swap USDC to Algorand

Powered by circle's bridge, we now offer wider options to users that want to interact in our pure non-custodial borrow/lend dapp. You can transfer USDC from any chain supported by circle bridge to any algorand address you want to.

The screenshot shows a 'Blockchain swap' interface. At the top right is a step indicator '1'. Below it, the text 'Blockchain swap' is displayed. The main content says 'You have 1 circle wallet' followed by a dropdown menu showing 'myWallet'. Below the dropdown are two buttons: a blue button labeled 'GENERATE DEPOSIT ADDRESS' and a smaller blue button labeled 'NEXT'.

- First select your circle wallet linked to an algorand address.

- Then generate a blockchain address that you'll deposit USD into and get it in the Algorand address linked to that



- Next send USD to the generated addresses

You will receive the deposited amount in your Algorand address soon. For testing purposes we recommend using smaller amounts.

For more, look into Technical review.

1.2 Neos interaction

1.2.1 Getting into Algorand world

Search for Algorand in the worlds menu.

1.2.2 Optin

To optin to DeFi4NFT in neos go through the shimmer.

1.2.3 Extract

Extract your Algorand NFT into neos world to use it as collateral.

1.2.4 Borrowing

Place an item on the top of the borrow table, specify the loan you want then touch the top of the table.

1.2.5 Repay

Place a USDC coin object onto the well, which represents repaying. State the amount you are going to repay and touch the well or click on repay label.

1.2.6 Burn

Get the dUSD representation near the fire, then state how much USDC you want to redeem. Then touch the torch stick(not the fire!) and confirm the transaction.

1.2.7 Portal

Click on the chain to bridge from. And send assets from the testnet of the chain. For ETH the network is Goerli.

1.3 Technical review

- NFTs used as collateral are frozen in account, only when account takes out loan.
- Frozen NFTs are unfrozen when full loan is paid back.
- There is no interest rate for borrowing USDCa.
- A 3% fee is paid to take out loan.
- Lenders sign a delegated logic signature to allow any account to withdraw USDCa that fulfill the following:
 1. Calls DeFi4NFT contract
 2. Withdraws atmost staked amount
- Any account that holds dUSD can claim 1:1 USDCa by sending the dUSD to DeFi4NFT contract.
- Borrower can borrow from upto 4 lenders
- Liquidation

How liquidation happens?

- Specify the addresss to liquidate
- Pay 95% of collateral's value to DeFi4NFT contract
- Set an account that will receive the liquidated asset
- You'll be sent the collateral to the address you specified

1.3.1 Smartcontract

There are three smartcontracts that power DeFi4NFT dapp.

1. DeFi4NFT Contract DeFi4NFT contract holds the state machine and locks/unlocks NFT in account (freezes/unfreezes NFT) . State machine, tracks:
 - `xids` tracks which NFT is used as collateral
 - `camt` tracks how much collateral is used for loan
 - `lamt` tracks how much loan is borrowed
 - `aamt` tracks how much loan is available from lender address
2. Liquidator Contract Liquidator contract reads current price of NFT from oracle and if loan is more than 90% of collateral it liquidates the NFT locked.
 - liquidator contract is the clawback address of leveragable NFTs on DeFi4NFT.
 - after liquidation completes the remainig asset is unfrozen. This is possible by AVM 1.1 (contract to contract call). Liquidator contract calls DeFi4NFT contract to unfreeze the asset.
3. Manager Contract Manager contract creates all other contracts on behalf of creator address. It also controls the NFTs that are configured to be borrowable in DeFi4NFT.

Notice

We have implemented here the Freeze admin and Clawback of algorand ASA. And IPFS for storing delegated LogicSig. We have also used Circle APIs for blockchain swap.

1.4 Install or build project

1.4.1 Building DeFi4NFT Locally

1.4.2 Create a sandbox environment

```
$ git clone https://github.com/Adg0/DeFi4NFT.git
$ cd DeFi4NFT
$ git clone https://github.com/algorand/sandbox.git
$ cd sandbox
$ ./sandbox up dev
```

Notice

Visit setting up [algorand sandbox](#) for development.

1.4.3 Create the contracts locally

```
$ cd cmd
$ go install .
$ ./cmd
```

1.5 TroubleShooting

1.5.1 CLI

If you encounter a failed transaction using the CLI, you can send the dryrun file to our contact or evaluate it your-self.

1.5.2 Web interface

Some common problem you might encounter is missing lenders when Borrowing.

- Solution is to search again, if the first search fails.

1.5.3 Neos

In VR you might need to scan QR-codes again. Especially for bridge, destroy the first instance and generate a second address if your non-Algorand wallet couldn't scan the address.

1.6 Additional features

1.6.1 Composibility

Since all contracts are written per Standard. You can extend the functionality.

One great thing about this project is its composibility. For instance, a liquidity provider at DeFi4NFT can extend the utility of their USDC by placing it in a liquidity pool and attaching a contract that will withdraw the asset from third party liquidity pool and lend it to a borrower at DeFi4NFT. In other words, they can get the best of both worlds.

- They have made a promise on DeFi4NFT, that means they will at somepoint get 3%
- They have kept the promise in a third party liquidity pool. That means they get some APY until it is actually lent out at DeFi4NFT. It is a profit made in two dapps with the same amount. We think this is gonna be a big incentive for liquidity providers.

1.7 Future

The next step for DeFi4NFT is to setup a voice command system so there would be more immersion for players.

1.8 Contact

Discord @1egen#0803

Discord @3spear#9556