Aave <> StarkNet Bridge

Massil Achab

Follow

Nethermind.eth

--

Listen

Share

by

[Massil Achab

](https://twitter.com/machilassab)

Thanks to

[Soufiane Hajazi

](https://twitter.com/quixotic_eth),

[Jorik Schellekens

](https://twitter.com/mempoolsurfer),and

[Swapnil Raj

](https://twitter.com/swp0x0)for helpful comments and revisions.

We are excited to announce the activation of the Aave <> StarkNet Bridge! The bridge lets you hold and trade Aave aTokens on StarkNet while still accruing Aave rewards from Ethereum. This collaboration between BGD labs and Nethermind marks the first step of Aave into the StarkNet ecosystem.

Ethereum <> StarkNet bridge for aTokens

A user can deposit their aTokens to the bridging contract on Ethereum. The bridge locks the aTokens, wraps them in staticATokens (explained below), and uses StarkNet's secure messaging mechanism to mint the equivalent amount of staticATokens on StarkNet. The cool thing about StarkNet messaging is that messages sent to StarkNet are verified as correct on Ethereum when the recipient StarkNet block is verified, rendering messages unforgeable!

What is a staticAToken?

The holder of an aToken earns their fees in the held aToken. As such, an aToken grows in balance. This constantly changing parameter is hard to bridge over continuously: staticATokens solve the bridging problem by no longer dynamically changing its internal state. A staticAToken represents the share of the aTokens a holder is entitled to, which stays constant. Working with staticATokens, we replace a continuously increasing number, their amount, with a constant number, the share, and avoid numerous updates between networks. Note traders and markets will still need to ascertain the staticAToken's actual value; however, this falls under the standard set of problems solved by oracles and arbitrage.

Reward accrual

StarkNet users will receive the same rewards as if they held their aTokens on Ethereum. When Aave's reward mechanism is active, the Aave protocol awards the bridge contract for the locked aTokens. Holders of staticATokens on StarkNet can claim these rewards according to the staticAToken's accounting of rewards. Aave rewards can be bridged back to Ethereum at any time by the holders. This way, rewards are fairly distributed proportionally to the time a user holds a staticAToken on the StarkNet side. Trading a token before claiming its accrued reward will result in a record of the outstanding reward due to the holder.

Governance

Governance still happens on Ethereum for this project. The Governance relayer can execute Governance actions on Ethereum on StarkNet. We use two Ethereum contracts from Aave and one StarkNet contract from the StarkNet DAI Bridge.

For instance, the execution of the activate_bridge spell, whose address is below, activated the bridge. The reader can find more details in this Ethereum-side AIP proposal.

Diagram

The following diagram provides an overview of the system and the interaction between the contracts.

How to bridge without coding

This section explains how to bridge Ethereum aTokens to StarkNet staticATokens using Etherscan UI and wallets on both networks.

Step 1 — Hold tokens that can be bridged

You need an Ethereum-compatible wallet funded with one of the six following tokens:DAI, aDAI, USDC, aUSDC, USDT, or aUSDT.

Step 2 — Get the tokens' balance

Select the amount you would like to bridge (note that the tokens above do not have the same number of decimals). To do this, go on the Etherscan page corresponding to your tokens, e.g., DAI, click on the tab "Read Contract" or "Read as Proxy," and call the function balanceOf

with your wallet address.

If you hold $50 of DAI / aDAI, the output of balanceOf

should be approximately 50000000000000000000

, and if you hold $50 of USDC / aUSDC / USDT / aUSDT, it should be about 50000000

.

Step 3 — Let the bridge transfer your tokens

Before depositing your tokens to the bridge, you should allow the bridge to spend your tokens. To do so, go on the token's Etherscan page, click on "Write Contract" or "Write as Proxy" tab, and then on "Connect to Web3" tab to connect your Ethereum wallet to Etherscan. Click on the function approve

, and fill in the bridge's address (0x25c0667E46a704AfCF5305B0A586CC24c171E94D

) as spender

and add the amount you would like to bridge as amount

. Finally, click "Write" and accept the transaction on your wallet.

Step 4 — Deposit your tokens to the bridge

To deposit tokens to the bridge, go to the bridge contract Etherscanpage. If your wallet is disconnected, click on "Connect to Web3" tab again, and click on the deposit

function to display its arguments. You should then enter the following inputs:

- l1AToken

: Fill in the address of the token you would like to bridge. For DAI, write 0x6B175474E89094C44Da98b954EedeAC495271d0F

.

- l2Recipient

: Fill in your StarkNet wallet address, converted to decimal. Usethis website, or BigInt

function in JavaScript to do this. For instance, if the StarkNet wallet address is 0x01270059Ea5843794F1130830800EcEF60B7D1AFd195f1847a884223a5B94f4A

, you should fill in 52122230822426253065445883306174534498450183722374412262861746209784236013 8

.

- amount

: Fill in the amount you would like to bridge. This amount should be lower or equal to the amount you have approved in the previous step.

- referralCode

: Fill in 0

. This argument is proper to identify future integrators.

- fromUnderlyingAsset

: Fill in false

if the token you bridge is an aToken (aDAI, aUSDC, aUSDT); otherwise, fill in true

.

Finally, click on “Write”, accept the transaction, and wait for Ethereum and StarkNet transactions to finish.

Step 5 — Import token to your StarkNet wallet

In your StarkNet wallet, click on “+ New token” for Argent X or “+ Add token” for Braavos, and fill in staticAToken’s address that corresponds to the tokens you have deposited on the Ethereum side — see the section below for deployed contracts’ addresses.

Deployment on Ethereum and StarkNet

Ethereum

- Bridge: proxy and implementation
- CrosschainForwarderStarknet: implementation
- AIP payload: implementation

StarkNet

- bridge: proxy and implementation class
- L2 governance relay: proxy and implementation class
- Static Aave v2 Ethereum aDAI: proxy and implementation class
- Static Aave v2 Ethereum aUSDC: proxy and implementation class
- Static Aave v2 Ethereum aUSDT: proxy and implementation class
- RewAave: proxy and implementation class
- activate_bridge spell: implementation class

Audits

Nethermind, Peckshield, and Certora audited the codebase. Please note that there is a Chinese wall between the Nethermind development team and the Nethermind audit team. We want to thank all the auditors for their meticulous work and remarks, which greatly improved the code.

Audit reports are available in the project repository here:https://github.com/aave-starknet-project/aave-starknet-bridge/tree/main/audit.

What’s next?

In the upcoming weeks, a discussion will start on the Aave governance forum to port the protocol Aave v3 to StarkNet, the project’s second phase. Please participate if you want this protocol to come to StarkNet.