

TL;DR;

As a follow-up of our [Debt Swap Adapter](#), we have released a new version [GHODebtSwapAdapter](#), now supporting users of Aave v3 Ethereum to swap any debt asset to vGHO (GHO variable borrowings).

Context

While our previously released Debt Swap Adapter (`ParaswapDebtSwapAdapter`

) automatically supports every newly added standard token on the Aave protocol, the custom nature of GHO requires an adaptation.

GHO has no underlying supply on the aToken, as GHO is minted when borrowing. This design is incompatible with the Aave style flash loans, as they use the underlying deposit as a liquidity source.

How to overcome the flash loans limitation?

While GHO does not support flash loans, in the [GHO listing proposal](#) by [@AaveLabs](#) a FlashMinter

Facilitator was introduced providing similar functionality but following the [EIP-3156](#) standard.

This mechanism allows sourcing “flash” GHO from a smart contract venue, even if not directly from Aave v3.

To account for the EIP-3156 flow, under the hood, GHO is handled separately in our new swap adapter. Different from the usual code path for a swap from e.g. vDAI

to vGHO

debt, the following steps will be performed:

1. Flash mint the amount of GHO

needed to repay your vDAI

debt

1. Swap the received GHO

to DAI

, in this case via Paraswap (to be compatible with previous versions).

1. Repay the vDAI

debt on behalf of the user.

1. Borrow enough GHO

to cover the “flash mint” + the fee of the facilitator (zero at the moment).

The adapter will continue to work exactly as it did before. There are 0 changes to the interface.

Why is it important?

Efficient debt swaps are a pretty important mechanism of arbitrage for stablecoins, as it allows users of protocols like Aave to change their exposure to more favorable interest rates, without any overhead of closing and re-open their position.

In a nascent stablecoin like GHO, this is especially important.

Next steps

- We have made the code public, so entities with user interfaces supporting Aave can start with the integration.

In the case of [app.aave.com](#) from [@AaveCompanies](#), the existing infrastructure for Repay with Collateral, Collateral Swap, and the v2→v3 migration we helped with, should make the integration straightforward.