

Summary

Introducing Aave v3.2, a new version of Aave V3 (currently v.3.1) including two important changes:

- A new feature: Liquid eModes
- The removal of all deprecated stable rate logic.

Context

July 27th, [Aave governance proposal 132](#) activated Aave v3.1 across all active networks. This currently active version of the Aave protocol was focused on security improvements (e.g. introducing virtual accounting), changing the architecture of the interest rate strategies, and multiple other updates to improve the operational efficiency of the system.

Additionally, with Aave v3.1 we introduced [Aave v3 Origin](#), a new repository for the Aave protocol (Foundry-based) to serve as the cornerstone to release further post-v3.1 versions faster while keeping maximum security standards.

One and a half months later, as a result of previous groundwork with v3.1 and Origin, we are glad to present to the community this new Aave v3.2 version.

Aave v3.2

The changes included in Aave v3.2 are more isolated than those of v3.1, and of different nature. While removing stable rate-related logic is more on the maintenance/codebase cleanup side, liquid eModes is a total user-facing and powerful feature.

1. Liquid eModes

Aave v3 introduced the concept of eMode (efficiency mode), an abstraction within the pool for configuring groups of assets with more capital-efficient risk configurations (LTV, Liquidation Threshold). This unlocked use cases like native high-leverage or even strategies between stablecoins. eMode could be deemed one of the first experiments in DeFi at scale with the concept of pools of assets within pools of assets.

The eMode feature has seen since then tremendous success, enabling use cases like ETH-correlated eMode, allowing for higher than default risk parameters on ETH LSTs/ETH positions (or MATIC, AVAX with their LSTs). However, the eMode feature always had some room for improvement, that way truly unlocking even more powerful use cases and flexibility on Aave.

A known technical limitation of the initial iteration of eMode is the following: a single eMode is optionally assigned to assets listed on Aave. That means that if let's say WETH is assigned eMode 1 eligibility (ETH-correlated, with ETH LSTs), WETH can't be eligible for any other eMode.

In different scenarios, this is an important constraint, as currently is not possible to have on the same Aave pool an eMode with wstETH and WETH, together with a different eMode with weETH and WETH, as WETH can only be in one of them at the same time.

The new Liquid eMode feature of Aave v3.2 removes the previous constraint: an asset listed on Aave can be eligible for different eModes, and then it only depends on the user to choose which eMode he wants to enter to

eMode, before and after v3.2

For example, with liquid eModes, a possible configuration not doable before would be:

- eMode 1, with wstETH, weETH and WETH.
- eMode 2, with wstETH and WETH.
- eMode 3, with weETH and WETH.
- eMode 4, with WETH and GHO.

So then, user A holding the wstETH and weETH collaterals, could borrow WETH at high LTV. User B holding only wstETH could borrow WETH at high (but different) LTV. User C holding only weETH could similarly borrow WETH at a different LTV than the previous two eModes. User D could have a position with WETH collateral and GHO borrowings.

This doesn't stop there, as more sophisticated configuration strategies could be adopted, like:

- eMode for only WETH and stablecoins.
- eMode for onboarding less mature LSTs, without requiring them being together with all major LSTs.

For extra configuration flexibility, liquid eModes also allow now to flag an asset as only borrowable, only collateral, or both, in the context of an eMode

. For example, in a hypothetic eMode with only wstETH and WETH, the normal configuration would be wstETH as only collateral and WETH as only borrowable, fully focusing on the wstETH leverage use-case.

In addition to the previous main characteristics of liquid eModes, we have also introduced the following minor changes to the system:

- Now the protocol doesn't enforce the LTV/LT of an asset being lower outside its eligible eModes. However, the validation gets stricter on the user side when entering any eMode (or from no-eMode): Health Factor is always validated to never allow the user to get his position below 1HF post-eMode swap.
- The protocol previously had an eMode-specific price feed, to be used to price with the same feed all assets on an eMode. However, this has never been used, and its utility (in its current form) is quite arguable. Therefore, we have to remove it from the data structure to allow more efficient storage allocation.
- On a technical note, before 3.2, the eMode of an asset was stored on the asset's data itself. Now, the eMode data stores all the necessary information for the protocol to understand which assets are eligible on that eMode.
- Similar as with previous upgrades on Aave v3, we have tried to follow a non-invasive approach when adding the feature, only touching/refactoring those pieces that were clearly necessary, but without doing any big high-level refactoring of the pool.

Security procedures

Similar as with any other Aave upgrade, we have applied multiple security measures that will be finished before the activation stage via governance proposal. Detailed information will be published once they are finalised.

2. Stable rate logic cleanup

Part of the v.3.1 upgrade was the addition of a function `swapToVariable()` for anybody to permissionless off-board stable borrowings to variable, that way factually deprecating completely stable rate on Aave pools.

With currently no active stable rate position ([moved via the Dolce Vita program by the service provider AC](#)), we have executed a follow-up step on the process: completely removing the stable rate logic from the protocol, which one hand simplifies the codebase, but on the other hand it also slightly optimises gas consumption and logic flows.

Even if fairly extensive change, this cleanup is non-invasive in how the protocol operates given that there are no active stable rate positions. All technical details about it will be included in the upcoming v3.2 release technical changelog.

Security procedures

Similar as with any other Aave upgrade, we have applied multiple security measures that will be finished before the activation stage via governance proposal. Detailed information will be published once they are finalised.

Next steps

1. Create an ARFC Snapshot for the upgrade of all Aave v3 instances to Aave v3.1.
2. In parallel, finish the preparations of the AIP payload for the upgrade, security procedures of v3.2 and publishing the results.
3. On-chain AIP creation, for the Aave governance to upgrade all pool instances across all networks.