TL;DR

Given the current market situation and past events (e.g. CRV bad debt), we want to start a discussion about the protocol's access control, permissions, and procedures for both Aave v2 and v3, in order to improve reaction speed under threats.

Opening for discussion the idea of having faster levers (e.g. Aave Guardian or some type of Risk Council) to change non-invasive risk configurations, in order to react swiftly under urgent circumstances.

Context

Currently, both Aave v2 and part of the v3 instances (those where possible) are controlled via the on-chain Aave governance system. This means that any type of change on assets, codebase, and parameters on each Aave liquidity pool instance, needs to go through an on-chain vote to be approved.

This is intended by design, in order to have a fully decentralized system on which AAVE holders have direct and factual control over everything in the ecosystem.

In practice, it is well known that this generalization over all types of permissions is not completely efficient, and the time delay created by governance (currently on Aave, minimum of ~5 days) sometimes creates a "lock" situation: a potential risk is unfolding, multiple people of the ecosystem know about it, but there is generally nothing to do.

As an example, having a "faster" mechanism on Aave v2 Ethereum would have allowed disabling borrowing on CRV, which almost surely would have reduced the bad debt to 0.

Even if this is more of a community ethos/governance/risk-management decision and not so much technical (our scope), from BGD we think it is important to initiate a discussion around the topic, especially making clear for the community that things can always be improved tech-wise if there is will for it.

Permissions on the Aave liquidity protocol

The system of access control/permission on Aave is not completely straightforward, due to the high customization/power of the protocol, and other aspects like having upgradeable components.

We will be releasing pretty soon a tool to give some extra transparency around all permissions of the Aave ecosystem, but for now, we will focus the discussion on the Aave v2 and Aave v3 sets of permissions, and what can potentially be improved in the really short term.

Aave v2

In what affects the Aave liquidity protocol, and slightly simplifying, the main smart contract managing access control on Aave v2 is the so-called PoolConfigurator, for example, the one for Aave v2 Ethereum HERE

Through this contract, the interactions allowed are:

- Doing the basic setup of listing a new asset: connecting the necessary smart contracts for the listing (a/v/s Tokens) and misc components like the incentives controller or the collector.
- Update implementations of aTokens, variable/stable debt Tokens. For example to add some new functionality or fix any bug.
- Enable/disable borrowing of an asset of the pool (also stable borrowing only).
- Change the collateral configurations of an asset. Liquidation Threshold, Liquidation Bonus and LTV.
- Activate/disable an asset. Meaning delisting; only possible if the activity on the asset is null.
- Freeze/unfreeze an asset. Meaning disabling supplying and borrowing liquidity of the asset, keeping all other actions available.
- Change the interest rate strategy of an asset. Meaning modifying the dynamics of borrow rates, like changing the so-called "slopes", and determining the acceleration of the rate after points of inflection of utilisation.
- Change the reserve factor of an asset. Percentage of the borrowing rate "redirected" to the Aave collector as fees of the protocol.
- Pause the whole pool. Not enabling any action, to be used only if a major threat affects the system, given that liquidation would be also disabled.

Apart from the permission on PoolConfigurator, it is also important to highlight the one to add/change oracle feeds for

assets, but generally, the holder of this is exactly the same as with the previous ones.

Currently, the split on who holds the permissions on v2 instances is pretty simple:

- An instance of the Aave Guardian (a multi-sig of elected community members) holds permission to pause the whole pool

. This is the entity assigned with EMERGENCY ADMIN

role.

- Everything else is controlled by the Aave governance Level 1 (short) executor, controlled via voting by all AAVE holders

. This is the entity assigned with POOL ADMIN

role.

- Only 1 entity can have the same role at the same time (but this can be potentially changed).

So for example, in the scenario of CRV, the options were:

1. pause() the whole protocol “fast” by mobilizing the Aave Guardian. Completely sub-optimal, as the consequences on all other assets are uncertain.
2. Pass a proposal to reduce somehow risk parameters of assets (e.g. CRV, USDC), disable borrowing, freeze, or any other measure.

Aave v3

Aave v3, as with almost everything else compared with v2, is a way more complex and customizable system, including granularity of permissions.

Interactions are still managed through a PoolConfigurator contract, but there are both more roles defining who has control over what, and it is possible to have multiple parties holding the same role.

Regarding interactions, v3’s are a superset of v2’s, with exactly the same levers as v2, plus:

- Enable/disable an asset to be borrowable in isolation mode.
- Set the debt ceiling for a collateral in isolation.
- Enable/disable an asset for siloed borrowing.
- Set new supply/borrow caps.
- Change the liquidation protocol fee, taken as a percentage of the liquidation bonus.
- Create a new eMode category.
- Add/remove an asset to/from an eMode category.
- Update the bridge protocol fee.
- Update the flash loan fees.
- Pause/unpause 1 asset, affecting all the positions containing it (different to v2, where the whole pool gets paused).

Currently, and simplifying the cross-chain governance aspects, the direction of the community is relatively similar (emergency admin for extreme measures, pool/risk admin for periodic ones), but with important possibilities already enabled on the codebase:

- The system has at the moment 4 roles, that can be given to multiple entities: EMERGENCY ADMIN

, POOL ADMIN

, RISK ADMIN

and ASSET LISTING ADMIN

.The functionality each one of them controls is also quite granular, and something that can be customized really easily by a technical party.

This granularity was introduced to 1) start with automation on smart contracts for different changes, giving temporarily/permanently the required role/s to them 2) as more professional parties onboard for contributions to the community, potentially having a “fast-track” procedure for limited functions (e.g. risk control).

A potential model. Example of Risk Council

As presented in the previous section, Aave v3 has already a quite powerful system of permissions in place, to adapt to any need of the community regarding the liquidity protocol. Aave v2 is behind in functionality, but we can affirm that adding some extra granularity of permissions is potentially doable if really required short term.

From our point of view, there is no step back decentralization-wise on having a fast mechanism (e.g. Aave Guardian or some different Risk Council formed by knowledgeable community members/entities) for certain procedures, especially if these are limited via smart contracts to only be able to execute actions that can’t really produce any harm in the system or its users (not affecting ever negatively their positions), but that can really make a big difference in threatening and urgent scenarios

.

Examples of those actions are:

- Set LTV to 0 on Aave v3. Factually reducing the “borrowing power” of a collateral asset to 0, but not affecting over-collateralization anyhow.
- Disable borrowing on an asset.
- Freezing an asset (disable supply and borrowing).

For the community to have a more precise example (only an example!

), this could look like the following:

- An Aave Risk Council

is formed via governance approval.

- 5 members, each one provably independent and without any conflict of interest, contributing only for the sake of a common good like the Aave protocol.
- A high understanding of Aave and DeFi is required, especially from an economic perspective.
- At least 2 members with also high understanding of the technical aspects of the protocol.
- Extremely high availability is required, with immediate replacement if not fulfilled.
- The Council would be represented on-chain with a multi-sig smart contract (e.g. Gnosis Safe). This multisig would receive the non-invasive permissions described in the previous section.
- To proceed with any action, 3-of-5 signatures are required. Always with justified reasons, even if not agreeing.
- By approving the Council initially, the Aave governance would authorize them to act on the limited set of actions at their discretion, with the goal of being as fast as possible under threat. If for security reasons the Council can’t disclose the actions in advance (highly probable), the full rationale of the decision should be disclosed whenever possible to do it in a responsible manner.
- Every 6 months, the Council’s performance is evaluated via on-chain governance, and potentially members are rotated.
- The ultimate goal is helping the community, so only really compromised and professional individuals and entities can be considered. Even if some type of compensation can be considered, it is probably wiser to look for members not motivated by it.
- At any point, the Aave governance has the power to remove all permissions from the Risk Council multi-sig. In addition, all the changes that can be executed by the Risk Council can be executed by the Aave governance too.

Next steps

Given that the potential effort on this is not really technical, but more on establishing a framework of procedures and defining which type of party would have special permissions, we request the community to participate in the discussion and propose different models, using the example of the Risk Council as a base.

From BGD, we will help with all technical aspects if a decision on this direction is taken.