

# TL;DR

Presenting to the community Aave Robot v1, the first iteration of a framework and architecture of smart contracts to automate permissionless actions of Aave, like those related to the Aave governance; powered by Chainlink Automation.

<https://github.com/bgd-labs/aave-governance-v2-robot>

## Context: Aave Governance v2 permissionless triggers

Given their nature, generally, blockchain systems don't incorporate any type of "automatic" triggering of actions: a smart contract always requires an external interaction (transaction) in order for anything to get executed.

Consequently, in some systems implementing so-called state machines, somebody needs to trigger with an external transaction the state transitions. For example, on Aave for a governance proposal to be queued, somebody needs to submit a transaction to directly or indirectly call the `queue()`

function.

It is important to clarify that these actions are totally permissionless, as any Ethereum address can trigger them, with the smart contract containing the logic to validate the state transition or other effects.

But in practice, certain "soft" problems appear as a consequence of the aforementioned:

- Anybody can call the permission-less functions, but more frequently than desired, only the entity behind let's say an Aave governance proposal has a rationale to spend gas doing it.

E.g. it has happened multiple times in the past that entities submitting governance proposals simply forget or are not familiar with extra steps post-submission, like queueing, execution, or cross-chain execution.

- Alerting infrastructure based on timing is not as friendly as it should be, and integrating it with all contributors to the protocol is more effort than it initially seems.
- There is a clear case to be made that if cost is scalable, a decentralized entity like the Aave protocol should cover the cost of its own purely operational workings.

Our conclusion is then: given the existing infrastructure out there and the integration effort, it is clearly worth it for Aave to try to automate all permissionless procedures

.

## Aave Robot: a layer of automation

The initial iteration of Aave Robot is simple, and based on the following:

- Focused on Aave Governance V2, which has the big majority of potential scope for automation.
- Set of smart contracts defining all the actions that can be triggered: queueing of proposals, execution, or cancellation.
- Funded by the Aave Governance itself with its holdings, and control at any point at which automation is running. This will be done via a `RobotOperator`

smart contract.

- Powered by Chainlink Automation, technology that we are currently more familiar with in terms of automation.

The following is a diagram of how Robot works in the case of Aave Governance V2:

And for the current Aave cross-chain governance:

This initial iteration supports the following:

- If the proposition power of a proposal creator drops during the pending/voting/queueing period below the required threshold, Robot will immediately cancel the proposal.
- Whenever the voting period (in blocks) of a proposal passes, Robot will automatically queue it.
- Whenever the timelock period (in seconds) of a proposal passes, Robot will automatically execute it. This applies both to the timelock on Ethereum and the timelock on the cross-chain executor of other networks.

Additionally, as commented the Aave Governance will have full control over Robot, by allowing:

- Registering new Chainlink automation as part of the Aave Ecosystem.
- Fund those automations both initially and by refilling afterward (this will also be possible to do for any other address, as a kind of “sponsorship”).
- Switch off automations, retrieving all funds to the Aave Collector.
- Change technical configurations, like for example the maximum gas limit to be accepted while running Robot. E.g. actions above 8'000'000 gas limit will need to be executed by the proposer, instead of from the DAO funds.

## And Aave Governance V3?

The upcoming Aave Governance V3 follows the same approach of Aave Governance V2: using permissionless state transitions, validated by the smart contracts themselves.

So even if the set of actions is bigger than in v2, the underlying automation needs are exactly the same, and naturally, Robot should be applied.

The actions requiring automation can be seen in the following diagram.

Additionally, with the cancel dynamics being similar to Governance v2, Robot will also cover the cancellation use case.

## Practical implementation and next steps

Given the permissionless nature of the project, we have been already testing and funding Aave Robot on Aave Governance v2 and cross-chain governance during the last 2 months, and now we think it is a good moment to fully integrate it into the DAO.

This means the following:

- We will create a Snapshot vote for the community to approve the integration.
- We will create and deploy RobotOperator

contracts that will act as a middleware for the DAO to have full control over which automation should be active and fund them.

- We will create a governance proposal for the DAO to fund (from Collector) all automation with the required LINK tokens. For context, during the last 2 months, ~270 LINK (close to 2'000 USD at current prices) has been spent on Ethereum Robot, with Polygon, Optimism, and Arbitrum with insignificant cost.

It is important to highlight that some proposals with high gas limits have been executed, which means they weigh importantly on the cost compared with all queuing and average executions.

- It is really important for the community to participate in the discussion on defining which kind of proposal executions should be included (and subsidized) by Aave Robot.

Some of the operational nature seems pretty clear that should be involved (e.g. risk parameters updates, caps, protocol upgrades, treasury management), but for example, new listings are something that could be reasonable to exclude (or not).