

# TL;DR

After the pre-approval of the community, with the development and security procedures completed, we present the practical steps for the activation of Aave Governance v3, migrating all Aave systems from Aave Governance v2.

The most important aspects:

- 2 migration governance proposals (running on v2)
- ) will simultaneously disable Aave Governance v2 and enable Aave Governance v3. The full process will take ~21 days.
- AAVE, stkAAVE, and aAAVE Ethereum as voting assets. The tokens will be upgraded and snapshots will be removed.
  - Polygon PoS and Avalanche C-Chain as initial voting networks (only 1 per proposal). Ethereum will be also enabled as a backup.
  - Voting will be accessible via the user interface [HERE](#) and multiple other venues.
  - All delegations will be reset just after the execution of the proposals.
  - All Aave cross-chain communication will be migrated to a.DI (Aave Delivery infrastructure).
  - All Aave infrastructure (no exception) will be governance controlled, with a.DI solving the previous technical limitations.
  - Initially, voting will be free for holders on Polygon and Avalanche: the Aave DAO will subsidize the aggregated voting cost, integrating Gelato.
  - The whole infrastructure will be automated with Aave Robot.
  - All Aave <> BGD tooling has been adapted to support Governance v3.

## Migration Governance v2→v3

Activating Aave Governance v3 requires passing both Level 1 (Short) and Level 2 (Long) proposals on Aave Governance v2, but as the execution of both should be atomic, the global procedure needs to be more sophisticated than usual.

The following diagram gives an overview of how this process will be.

Key points:

- Mediator contract
- . In order to execute sync/atomically both Level 1 and Level 2 proposals, we have created the so-called mediator smart contract, which will receive permissions from the Level 2 executor (long), allowing for the Level 1 executor to do its own execution in sync.

This is mandatory, because in any situation of Level 1 passing and Level 2 not (or vice versa), the whole Aave Governance v2 & v3 systems will not be operative.

- No new governance proposal on Level 1 can be created after the creation of our Level 1 migration proposal.

This is mandatory, as as soon our Level 1 proposal executes via Mediator, Governance v2 becomes immediately non-functional, and with it, any pending proposal.

- By creating the Level 1 proposal days after Level 2 was running (and potentially in timelock already), it will be possible to maximize the time available to create other proposals for the community. Again, all of them should be scheduled to finish (executed, canceled, failed) before the execution of our Level 1.
- Step-by-step, the timeline will be as follows:
  - Day 0.
  - Level 2 proposal (Long) is created.
  - Level 2 proposal (Long) is created.
  - Day 1.

- Delay of Level 2 proposal finishes, voting starts
- Delay of Level 2 proposal finishes, voting starts
- Day 11.
- Voting of Level 2 proposal finishes, enters timelock of 7 days.
- Voting of Level 2 proposal finishes, enters timelock of 7 days.
- Day 14.
- Level 1 proposal (Short) is created. After this, no other governance proposal should be created until the whole process is finished.
- Level 1 proposal (Short) is created. After this, no other governance proposal should be created until the whole process is finished.
- Day 15.
- Delay of Level 1 proposal finishes, voting starts.
- Delay of Level 1 proposal finishes, voting starts.
- Day 18.
- Timelock of Level 2 proposal finishes and executes. It sends all permissions to the Mediator contract, waiting for Level 1 execution
- Voting of Level 1 proposal finishes, enters timelock of 1 day.
- Timelock of Level 2 proposal finishes and executes. It sends all permissions to the Mediator contract, waiting for Level 1 execution
- Voting of Level 1 proposal finishes, enters timelock of 1 day.
- Day 19.
- Level 1 proposal executes, moving all the Level 1 permissions to Governance v3. Additionally, it calls the Mediator to move all the Level 2 permissions from there, to Governance v3.
- As the Mediator now has permissions of both Level 1 and Level 2 Executors, it enqueues and immediately executes both "real" payloads.
- L2 payloads are time-locked on their respective networks.
- Aave Governance v2 becomes inoperative.
- Aave Governance v3 becomes operative.
- Level 1 proposal executes, moving all the Level 1 permissions to Governance v3. Additionally, it calls the Mediator to move all the Level 2 permissions from there, to Governance v3.
- As the Mediator now has permissions of both Level 1 and Level 2 Executors, it enqueues and immediately executes both "real" payloads.
- L2 payloads are time-locked on their respective networks.
- Aave Governance v2 becomes inoperative.
- Aave Governance v3 becomes operative.
- Day 21.
- Execution of L2 payloads on all non-Ethereum networks.
- Execution of L2 payloads on all non-Ethereum networks.
- Day 0.
- Level 2 proposal (Long) is created.
- Level 2 proposal (Long) is created.

- Day 1.
- Delay of Level 2 proposal finishes, voting starts
- Delay of Level 2 proposal finishes, voting starts
- Day 11.
- Voting of Level 2 proposal finishes, enters timelock of 7 days.
- Voting of Level 2 proposal finishes, enters timelock of 7 days.
- Day 14.
- Level 1 proposal (Short) is created. After this, no other governance proposal should be created until the whole process is finished.
- Level 1 proposal (Short) is created. After this, no other governance proposal should be created until the whole process is finished.
- Day 15.
- Delay of Level 1 proposal finishes, voting starts.
- Delay of Level 1 proposal finishes, voting starts.
- Day 18.
- Timelock of Level 2 proposal finishes and executes. It sends all permissions to the Mediator contract, waiting for Level 1 execution
- Voting of Level 1 proposal finishes, enters timelock of 1 day.
- Timelock of Level 2 proposal finishes and executes. It sends all permissions to the Mediator contract, waiting for Level 1 execution
- Voting of Level 1 proposal finishes, enters timelock of 1 day.
- Day 19.
- Level 1 proposal executes, moving all the Level 1 permissions to Governance v3. Additionally, it calls the Mediator to move all the Level 2 permissions from there, to Governance v3.
- As the Mediator now has permissions of both Level 1 and Level 2 Executors, it enqueues and immediately executes both “real” payloads.
- L2 payloads are time-locked on their respective networks.
- Aave Governance v2 becomes inoperative.
- Aave Governance v3 becomes operative.
- Level 1 proposal executes, moving all the Level 1 permissions to Governance v3. Additionally, it calls the Mediator to move all the Level 2 permissions from there, to Governance v3.
- As the Mediator now has permissions of both Level 1 and Level 2 Executors, it enqueues and immediately executes both “real” payloads.
- L2 payloads are time-locked on their respective networks.
- Aave Governance v2 becomes inoperative.
- Aave Governance v3 becomes operative.
- Day 21.
- Execution of L2 payloads on all non-Ethereum networks.
- Execution of L2 payloads on all non-Ethereum networks.

The independent actions included in the proposal are the following:

Mediator component

- The Mediator contains 2 entry points: the first for the actions to be executed once Level 1 is ready (Level 2 being ready before); the other for cancellation/rollback of permissions, if any of the proposals will not pass. Regarding the “execution” actions:
- Upgrade the implementations of AAVE and stkAAVE.
- Transfer the ownership of the ProxyAdmin Level 2 to the Governance v3 Level 2 Executor.
- Call the Governance v3 Level 2 Executor to accept the ownership of the Governance v2 Level Executor.
- Transfer the ownership of the Governance Level 2 Executor to the Governance v3 Payloads controller.
- Upgrade the implementations of AAVE and stkAAVE.
- Transfer the ownership of the ProxyAdmin Level 2 to the Governance v3 Level 2 Executor.
- Call the Governance v3 Level 2 Executor to accept the ownership of the Governance v2 Level Executor.
- Transfer the ownership of the Governance Level 2 Executor to the Governance v3 Payloads controller.
- On the side of cancellation/rollback (callable by Guardian at any time, or by any address if a grace period has passed since Level 2 was executed):
- Transfer back the ownership of the ProxyAdmin Level 2 to the Governance v2 Level 2 Executor.
- Transfer the ownership of the Governance v3 Level 2 Executor back to the Governance v2 Level Executor (from the Mediator itself).
- Transfer back the ownership of the ProxyAdmin Level 2 to the Governance v2 Level 2 Executor.
- Transfer the ownership of the Governance v3 Level 2 Executor back to the Governance v2 Level Executor (from the Mediator itself).

#### Level 1 (Short executor)

##### Ethereum

- Trigger the execution of the Mediator.
- Fund CCC contract on a.DI from Aave Collector.
- Fund execution Robot.
- Migrate all “minor” permissions of stkAAVE to the Governance v3 Level 1 Executor.
- Migrate all GHO permissions to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v1 Ethereum to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v2 Ethereum to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v2 Ethereum AMM to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v3 Ethereum to the Governance v3 Level 1 Executor.
- Migrate permissions of Aave MerkleDistributor (rescue mission) to the Governance v3 Level 1 Executor.
- Migrate admin of LendToAaveMigrator to the Governance v3 Level 1 Executor.
- Migrate permissions of Balancer v1 ABPT to the Governance v3 Level 1 Executor.
- Transfer the ownership of the Aave Swapper contract to the Governance v3 Level 1 Executor.
- Transfer the ownership of the Aave Collector Polygon→Ethereum bridge to the Governance v3 level 1 Executor.
- Migrate the admin role of Governance v2 Level 1 Executor to the Governance v3 Level 1 Executor.
- Change ownership of Governance v3 Level 1 Executor to the Governance v2 Payloads Controller.

##### Polygon PoS

- Fund Gelato for Aave gas tank.
- Fund CCC contract on a.DI from Aave Collector.

- Fund execution Robot.
- Migrate all permissions of Aave v2 Polygon to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v3 Polygon to the Governance v3 Level 1 Executor.
- Transfer the ownership of the Aave Merkle Distributor contract to the Governance v3 Level 1 Executor.

#### Arbitrum

- Fund execution Robot.
- Migrate all permissions of Aave v3 Arbitrum to the Governance v3 Level 1 Executor.

#### Optimism

- Fund execution Robot.
- Migrate all permissions of Aave v3 Optimism to the Governance v3 Level 1 Executor.
- Transfer the ownership of the Aave Merkle Distributor contract to the Governance v3 Level 1 Executor.

#### Metis

- Migrate all permissions of Aave v3 Metis to the Governance v3 Level 1 Executor.

#### Base

- Migrate all permissions of Aave v3 Base to the Governance v3 Level 1 Executor.

#### Guardian proposal (Avalanche)

- Fund CCC contract on a.DI from Aave Collector.
- Migrate all permissions of Aave v2 Avalanche to the Governance v3 Level 1 Executor.
- Migrate all permissions of Aave v3 Avalanche to the Governance v3 Level 1 Executor.
- Migrate permissions of Proof of Reserve to the Governance v3 Level 1 Executor.

#### Level 2 (Long executor)

- Transfer the proxy admin permissions from the AAVE token to the ProxyAdmin Level 2 contract.
- Transfer the ownership of the ProxyAdmin Level 2 contract to the Mediator.
- Transfer the pendingAdmin role of Governance v2 Level 2 Executor to the Governance v3 Level 2 Executor.
- Change ownership of Governance v3 Level 2 Executor to the Mediator contract.

One really important step security-wise is the ownership change of previous Executors/BridgeExecutor to the new v3 Executors. This provides full assurance that no permission will be locked during the migration

.

## General implications of the upgrade

Even if pretty transparent for end users, most of the Aave smart contracts infrastructure will be affected by the activation of Aave Governance v3, as de-facto, all-controlled systems (and how they are controlled) will change.

The following is a list of systems impacted:

- Aave Governance v2
- . All smart contracts will not be used anymore, meaning that the entry point for voting or any other interaction will change.
- Aave permissions

. Currently, on Ethereum, all Aave permissions are held by the Aave Governance v2 Executor contracts (Level 1 and Level 2), and on other networks, by the BridgeExecutors (Polygon, Optimism, Arbitrum, Metis, Base).

As previously described, the activation proposal of Governance v3 will migrate all the permissions to new Executor contracts

on all networks.

- Governance user interface

. Technically, the new governance system works differently from v2. That's the reason why we have developed a new Aave Governance V3 user interface, for participants to interact with the permissionless smart contracts of v3.

All other community interfaces will stop working once the activation proposal passes and will need to be adapted to the new system.

- Aave cross-chain governance

. As mentioned on permissions, now all networks will share the same pattern of identical Executors, holding all Aave permissions.

This is a consequence of the re-architecture of the proposals' lifecycle: now, everything (Ethereum or non-Ethereum) was designed holistically, not incrementally.

Additionally, under the hood, the whole system is powered by a.DI (Aave Delivery Infrastructure).

After the upgrade, the current Aave cross-chain governance will not be operative anymore.

- Voting assets' implementations (AAVE, aAAVE, stkAAVE)

. The contracts have been adapted and optimized to work with the Aave Governance v3 system, mainly to reduce gas costs, and be storage-proof friendly.

The addresses of the tokens will stay the same

.

## Changes in governance flows

Even if we tried to stay as compatible as possible with the v2 system, the re-architecture and addition of new components/features naturally affect certain flows of participation in the Aave governance.

### How to access Aave Governance v3?

Aave Governance v3 is a permissionless system, with users participating in it by directly submitting transactions to smart contracts.

But for convenience and good user experience, we have developed a user interface application and additional tooling that helps participants visualize data and easily build transactions.

Via the user interface

The user interface can be used in the following ways:

- Running it locally

. Everybody can download the UI codebase, introduce its credentials (e.g. RPC URL, IPFS gateway), and run it in a local environment. This process is straightforward, with all the instructions [HERE](#).

- Via the decentralized IPFS deployment

. Another option is to access the IPFS deployment, which doesn't require any credentials, as it runs in completely public infrastructure.

Given the nature of IPFS, the most updated links will change as different versions get deployed and can be tracked [HERE](#) (link on the "open" icon to the right of the last commit).

- Via [vote.onaave.com](https://vote.onaave.com)

. Hosted version of the interface by BGD Labs, same as the other access points, just with some speed optimization.

The final version will be active just after the migration v2→v3 is executed.

Whenever opened the first time (or afterward via the top bar), an interactive Tutorial can be used to learn about all available actions.

Via a block explorer (Polygonscan, Snowstrace, Etherscan)

Another way of voting is to use a web browser together with a block explorer. The function that needs to be called is `submitVote()`

on the VotingMachine

of the corresponding proposal's voting network.

As constructing the storage proofs required as input can be pretty challenging for non-technical users, we have added a utility to `aave-cli` that generates the input parameters in the format expected by block explorers.

All the instructions on how to use this tool can be found on the [README](#).

Via a Foundry script

If you prefer to use a hardware wallet like Ledger or (not so recommended) a private key, without touching the browser, we have also created a Foundry script that anybody can run in their machine to emit a vote.

All the instructions on how to use it can be found on [HERE](#).

## Delegation

In general, the dynamics of delegation of voting/proposition power remain the same as on v2: you can delegate your power per voting asset to any Ethereum address, who can use it to vote on your behalf.

Apart from AAVE and `stkAAVE`, now `aAAVE v3` Ethereum will hold voting power too.

For users interacting directly with the smart contracts, delegations are managed exactly the same as on v2, by calling the appropriate functions on each token contract (AAVE, `stkAAVE`, `aAAVE`).

Some differences from the v2 system:

- The delegations management should be done on the new user interface. The application has an interactive FAQ showing step-by-step how to do it.
- Current delegations will be reset at the moment of the proposal execution (day 19), so delegators and delegates should be in touch to set them up again

## Voting

Voting is the most impacted flow after the migrations.

On Governance v3, whenever a proposal is created, a Voting Network is selected, for example, Polygon PoS for proposal 350.

For users interacting directly with smart contracts, voting should be done by calling the appropriate functions on the VotingMachine

deployed in the Voting Network of the corresponding proposal. So on the Aave VotingMachine

of Polygon PoS in the example of proposal 350.

Instructions on how to submit a vote can be found in the previous section.

## Voting representatives

As we commented previously, on v3 a vote consists of a choice (YES, NO) and proof of voting power. This proof of the voting balance is intrinsically linked to the address emitting the vote (the signer of the vote transaction), and this can create the following problematic situation:

- A multi-sig A (e.g. Gnosis Safe) is receiving voting power delegation on Ethereum.
- As the multi-sig is a smart contract, it can't sign transactions, which means that nobody can submit a vote on its behalf on Polygon, using the normal meta-transaction vote.
- Also, the multi-sig in this case only exists on Ethereum, and not on Polygon, and there is no procedure to re-create it in exactly the same address on Polygon (possible, but currently not general practice).

As this can be a relatively common situation, we have introduced the concept of representative of a voter

, which will work the following way on the previous example:

- Multi-sig A calls the Governance contract on Ethereum to choose a Polygon address B as its representative on Polygon.
- This means that B will be able to vote on behalf of A, for all proposals happening on Polygon, with the power of A (both balance and delegations).
- In practice, the members of multi-sig A can create another multi-sig with the same members on Polygon, choosing it as a representative, and do vote transactions from the multi-sig of Polygon from then on.

To choose a representative, an address in Ethereum should call the `updateRepresentativesPerChain()`

function on the GovernanceCore

smart contract.

As an alternative to the smart contracts flow, the Aave Governance v3 interface also has the functionality to both choose a representative and switch between addresses you can represent (as one address can be representative of multiple, including itself).

## Creation and testing of proposals

We have adapted all the Aave <> BGD tooling to be compatible with Aave Governance v3, making the transition smooth for everybody already using tools like Aave Seatbelt, Aave helpers, and especially, Aave Proposals.

On Aave Proposals, the changes should be pretty transparent to all users, as the proposals' generator CLI (`yarn generate`) will automatically support Governance v3 once it is active.

## Cancellation fee

Something new introduced on Aave Governance v3 is the concept of a cancellation fee

. On this new version, the assets carrying proposition power don't have snapshots anymore, so there could exist a spam vector of creation of proposals, which would cause Aave Robot to spend funds automatically canceling them.

To protect against that, now on proposal creation, the cancellation fee (0.05) ETH is required. If the proposal gets canceled anyhow, the cancellation fee is used to cover the cost of Aave Robot. However, if the proposal finishes in any other state (passes, fails), any address can claim back the cancellation fee from the governance smart contract, which will be sent to the proposal creator.

# a.DI (Aave Delivery Infrastructure)

(Introduction to a.DI [HERE](#))

Even if a completely separate system in nature, a.DI (Aave Delivery Infrastructure) will be activated in parallel with Aave Governance v3, as it powers all the cross-chain communication on the proposals lifecycle.

This means that all Aave systems will be fully controlled by the Aave Governance

.

The a.DI initial specifications by communication channels will be the following:

- Ethereum ↔ Polygon PoS.
- Consensus: 3-of-4.
- Bridge providers: Chainlink CCIP, LayerZero, Hyperlane, and native bridge.
- Chainlink Emergency Mode Oracle: Enabled
- Consensus: 3-of-4.
- Bridge providers: Chainlink CCIP, LayerZero, Hyperlane, and native bridge.
- Chainlink Emergency Mode Oracle: Enabled
- Ethereum ↔ Avalanche C-Chain.
- Consensus: 2-of-3



- Bridge providers: Chainlink CCIP, LayerZero, Hyperlane.
- Chainlink Emergency Mode Oracle: Enabled
- Consensus: 2-of-3
- Bridge providers: Chainlink CCIP, LayerZero, Hyperlane.
- Chainlink Emergency Mode Oracle: Enabled
- Ethereum → Optimism.
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Ethereum → Arbitrum.
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Ethereum → Base.
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Ethereum → Metis.
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled
- Consensus: 1-of-1
- Chainlink Emergency Mode Oracle: Disabled

Initially, only the Aave Governance v3 contracts will have permission to send messages via a.DI

.

Regarding permissionless, a.DI is a governance-controlled system from day 0 for higher-order permissions (e.g. upgradeability, addition/removal of bridge providers, modification of consensus rules, or guardian replacement).

For lower-order permissions:

- Given that the goal of the retry mechanism is precisely to avoid on-chain proposals overhead just to retry, the permissions to do it if there is any downtime on underlying infrastructure will be held by BGD Labs.
- If the Aave Governance on Ethereum activates at any point the Chainlink Emergency Mode, the Aave Guardian will get permission to rescue the system.

All the specific details about a.DI (addresses, general documentation) can be found [HERE](#).

## Aave Robot

(Introduction to Aave Robot [HERE](#))

Aave Robot is the automation layer of all Aave actions, powered under the hood by Chainlink Automation.

Currently, Aave Robot is already [running and controlled by the Aave DAO for Aave Governance v2](#) but with Aave Governance v3, the scope of Robot gets expanded, and will cover the following automations:

- Activate voting (Ethereum)
- . After the delay post-proposal-creation passes, Robot will snapshot the block on Ethereum which will determine voting power, and send it to the Voting Network via a.DI.
- Settlement voting global data (Voting Network)
- . Once the block of the proposal arrives at the Voting Network from Ethereum, Robot will “settle” global information required for voting, by extracting it from the block via storage proofs.
- Create a vote (Voting Network)
- . Once both the block and its required data are settled on the Voting Network, the Robot will trigger the start of the voting period.
- Close and send a vote (Voting Network)
- . Once the voting period is finished on the Voting Network, Robot will trigger its closure and send the results back to Ethereum for validation.
- Execute proposal (Ethereum)
- . Once the results have arrived on Ethereum, Robot will trigger the proposal execution, which in the context of Core Network means sending to the different timelocks the payloads.
- Execute payload (Each of the Execution Networks)
- . Once the timelock period finishes on any PayloadsController
- , the Robot will trigger the final execution.
- Cancel proposal (Ethereum)
- . If any of the cancellation conditions are fulfilled during the lifecycle of a proposal, Robot will trigger the cancellation.
- It is important to highlight that all the actions executed by Robot are totally permission-less (can be executed by any address). Still, Robot removes all the monitoring and operational overhead of executing them by third parties.

## Aave Gelato gas tank

While designing Aave Governance v3, our initial goal was to make voting as cheap as possible, by moving its processing to a different environment than Ethereum.

Soon after, looking at the gas costs of the future Voting Networks, we realized we could take the idea forward and enable voters to participate for free, just by signing messages and the Aave DAO paying the aggregated cost.

Building relayer infrastructure is no trivial task, so we contacted a long-term partner of Aave like Gelato to understand how we could integrate their system into our needs, and the result is the Aave <> Gelato gas tank.

The idea behind the Aave <> Gelato gas tank is simple:

- When voting, instead of submitting a transaction to the Voting Network (e.g. Polygon PoS), the user will sign the vote and send it to the Gelato infrastructure. This means the user doesn't need to pay any cost for gas.
- If it is a valid vote, Gelato relayers infrastructure will submit the transaction on behalf of the user, by using the meta-transactions capabilities of Aave Governance v3.
- The Aave DAO will cover the voting cost by refilling (with any supported currency) a “gas tank” address, from where Gelato will withdraw funds periodically.
- Voting via Gelato is totally optional, not compromising anyhow decentralization.

To start with, the governance proposal activating Governance v3 will transfer 10,000 aUSDC from v3 Polygon to the Gelato, in order to cover all voting costs for approximately 20/30 proposals

# Licensing of the codebases

Aave Governance v3, a.DI (Aave Delivery Infrastructure), and Aave Robot are pretty valuable projects for the Aave DAO.

Given their criticality, and that we produce them as part of the service we provide, we think it is up to the Aave DAO to decide their licensing. But in order to publish the code and at the same time keep IP protection, we have released all the core repositories under BUSL1.1 license, not permitting any entity trying to harm or compete with Aave to use them anyhow.

However, the following considerations are really important:

- It is completely up to the DAO to make a decision to change the license/s at any point, to fully open sources GPL compatible alternatives (e.g. MIT). For this, the Aave governance contracts can just modify the change-date

record on the different subdomains of aavelicense.eth

(we have transferred its ownership to the DAO.

- Apart from requiring attribution, the current BUSL1.1 allows anybody not competing with Aave to use, fork, or modify the codebases, if this doesn't break the mandatory clauses.

More information can be found on any of the licenses, for example [HERE](#).

## FAQ

- Which voting networks will be available to start with?
- Polygon PoS and Avalanche. Roll-up environments will probably come pretty soon, depending on some extra research on the a.DI side.

Ethereum is also enabled, but only as last-resort backup.

- Polygon PoS and Avalanche. Roll-up environments will probably come pretty soon, depending on some extra research on the a.DI side.

Ethereum is also enabled, but only as last-resort backup.

- How to choose a Voting Network for a proposal?
- It is up to the proposal creator to choose where his proposal will be voted. However, the cost of voting is the most important factor to consider.

For the first 5-10 proposals, we recommend alternating between Polygon PoS and Avalanche, in order to test the systems and understand better costs in production.

- It is up to the proposal creator to choose where his proposal will be voted. However, the cost of voting is the most important factor to consider.

For the first 5-10 proposals, we recommend alternating between Polygon PoS and Avalanche, in order to test the systems and understand better costs in production.

- To recap, how different the system is in the duration of states (post-creation, voting, timelock, etc.) compared with Aave Governance v2?
- Generally, the timings are exactly the same as on v2.
- Cooldown post-creation

: same as in v2, 1 day.

- Voting period

: same as in v2, 3 days (or 10 for Level 2).

- Time-lock after proposal passed

: same as in v2, 1 day on Ethereum (7 days for Level 2), 2 extra days whenever the payload needs to be executed on other Execution networks.

- Cooldown post-creation

: same as in v2, 1 day.

- Voting period

: same as in v2, 3 days (or 10 for Level 2).

- Time-lock after proposal passed

: same as in v2, 1 day on Ethereum (7 days for Level 2), 2 extra days whenever the payload needs to be executed on other Execution networks.

- Given the cross-chain communication (a.DI) and automation (Robot) involved, we expect to have some extra minor delay between the different states, which should not amount to more than 2 aggregated hours per proposal.
- Generally, the timings are exactly the same as on v2.
- Cooldown post-creation

: same as in v2, 1 day.

- Voting period

: same as in v2, 3 days (or 10 for Level 2).

- Time-lock after proposal passed

: same as in v2, 1 day on Ethereum (7 days for Level 2), 2 extra days whenever the payload needs to be executed on other Execution networks.

- Cooldown post-creation

: same as in v2, 1 day.

- Voting period

: same as in v2, 3 days (or 10 for Level 2).

- Time-lock after proposal passed

: same as in v2, 1 day on Ethereum (7 days for Level 2), 2 extra days whenever the payload needs to be executed on other Execution networks.

- Given the cross-chain communication (a.DI) and automation (Robot) involved, we expect to have some extra minor delay between the different states, which should not amount to more than 2 aggregated hours per proposal.
- I only trust smart contracts and code I run by myself. How can I emit votes by running all the infrastructure by myself? (no UI, no API, only code I can read and run).
- Option 1. Run locally the user interface following the instructions [HERE](#).
- Option 2. Deploy your own hosted version on Vercel. Instructions will be added in the repository during the following days.
- Option 3. Run locally aave-cli, to get data about proposals or build vote transactions, that later on can be submitted via a block explorer. All instructions can be found [HERE](#).
- Option 4. Vote via the forge script explained [HERE](#).
- Option 1. Run locally the user interface following the instructions [HERE](#).
- Option 2. Deploy your own hosted version on Vercel. Instructions will be added in the repository during the following days.
- Option 3. Run locally aave-cli, to get data about proposals or build vote transactions, that later on can be submitted via a block explorer. All instructions can be found [HERE](#).
- Option 4. Vote via the forge script explained [HERE](#).
- Under which conditions a proposal can be canceled?
- If proposition power is below the required threshold at any point during the pre-execution lifecycle of a proposal.
- Guardian can cancel proposals, as a protection against malicious ones.

- The proposal creator can cancel his own proposals (new on v3).
- If proposition power is below the required threshold at any point during the pre-execution lifecycle of a proposal.
- Guardian can cancel proposals, as a protection against malicious ones.
- The proposal creator can cancel his own proposals (new on v3).

## Links

- Aave Governance V3 smart contracts  
<https://github.com/bgd-labs/aave-governance-v3>
- Aave Governance V3 interface  
<https://github.com/bgd-labs/aave-governance-v3-interface>
- a.DI (Aave Delivery Infrastructure)  
<https://github.com/bgd-labs/aave-delivery-infrastructure>
- Aave Robot v3  
<https://github.com/bgd-labs/aave-governance-v3-robot>
- Governance v2→v3 migration proposals  
<https://github.com/bgd-labs/gov-v2-v3-migration>
- AAVE token v3  
<https://github.com/bgd-labs/aave-token-v3>
- aAAVE governance v3 compatible  
<https://github.com/bgd-labs/aave-a-token-with-delegation>
- stkAAVE governance v3 compatible  
<https://github.com/bgd-labs/aave-stk-gov-v3>
- Security procedures
- All repositories contain a list of security procedures applied. Props to all the parties involved: [Emanuele Ricci](#), [Certora](#) and [SigmaPrime @sigp](#).
- All repositories contain a list of security procedures applied. Props to all the parties involved: [Emanuele Ricci](#), [Certora](#) and [SigmaPrime @sigp](#).

## Next steps

After some days of discussion, we will proceed with the previously outlined activation plan of Governance v3.