

# Autonolas, Autonolas-Governance, Autonolas-ARC, and Autonolas-AIP

## Overview of the project and its mission

Autonolas' mission is to provide a **foundation** on which new types of autonomous applications can be built. These applications can enable complex processing, take action on their own, run continuously, and are crypto-native, that is decentralized, trust-minimized, transparent, and robust. We also call these applications *Agent Services*, as they are primarily built as a multi-agent-system that is run off-chain.

The logic of an agent service is distributed among the participating agents, and the service intermediate states are replicated on a so-called consensus gadget (a sort of short-lived blockchain). The most important intermediate values obtained in a given service are secured on a public blockchain. Which values are committed to the public blockchain depend on the agent service, the design of which is decided by the service owner.

To operate agent services and achieve full-stack autonomy, Autonolas will provide the following:

- a. An open-source software stack to realize these services
- b. An on-chain protocol, eventually deployed on all major smart-contract blockchains, that secures these services and incentivizes their creation.
- c. A tokenomics model resulting in a virtuous flywheel designed to attract more and more capital and developers, in turn increasing the number of autonomous applications.

The Autonolas stack places software composability at the core of its design. Composability at the stack and services layers enables Autonolas to convert linear component and service growth into exponential application opportunities.

## Governance

Governance is particularly important in a modular system as it is used to vote on the adoption or abandoning of modes of functioning. Autonolas' governance system is designed to assume various control points, such as launching and managing Agent

Services (see the so-called [Protocol-owned Services]([[https://twitter.com/autonolas/status/1504120575849074699?t=O2cRu-X8yVnqDFm\\_Qu8jNg&s=19](https://twitter.com/autonolas/status/1504120575849074699?t=O2cRu-X8yVnqDFm_Qu8jNg&s=19)]) for more details), managing minting rules for contributed software components, and upgrading various parameters in the tokenomics.

Building on the experience of existing decentralized protocols, such as Compound, we created three distinct components for governance: the veOLAS virtualized-token (locked claim on OLAS), the GovernorOLAS governance module, and the Timelock. Together, these components will allow the community to propose, vote, and implement changes. Proposals can notably modify system parameters, support new technological directions, or add entirely new functionality to the protocol.

## OLAS token

OLAS utility token of Autonolas protocol that follows the ERC20 standard. Initially, OLAS can be used for locking to participate in governance. The purposes of OLAS increase when the other parts of the Autonolas protocol will be deployed. In particular, OLAS will allow the protocol to acquire assets, including their own liquidity, in exchange for OLAS tokens at a discount and, as a byproduct, will attract more developers in contributing to the protocol.

## veOLAS

The name veOLAS stands for voting escrowed OLAS and it is built upon the experience of existing decentralized protocols, such as Curve finance. veOLAS is a non-tradeable virtualized token that can be obtained by locking OLAS. Users that want to participate in Autonolas governance are required to lock some of their OLAS tokens for a maximum of four years. veOLAS are used then to determine the voting power of an account and this voting power decays linearly as the remaining time of OLAS unlock decreases.

## The Governance process

Autonolas DAO will be made up of holders of veOLAS which have non-delegable voting power. Any address that holds at least 5000 veOLAS can create a governance proposal that can be voted upon by voters.

When a governance proposal is created, it enters a two days **review period**, after which voting begins. Voting lasts for three **election duration** days and any address with veOLAS can vote *for* or *against* or *abstain* the proposal. If a majority and at least 3% of the global voting power are cast *for* the proposal, the latter is queued in the Timelock and can be implemented two days later.

Exceptionally, some changes to the Autonolas Protocol could be executed by a **community-owned multisig wallet**, bypassing the governance process. This allows a set of trusted actors to overrule governance in certain aspects, e.g. security exploits that need to be patched without governance discussion.

### Off-chain governance

Optionally, the on-chain proposal stage can be preceded by an off-chain signaling vote, using the [Snapshot](#) platform. Off-chain signaling can be a good tool to gauge the interest of community participants with low willingness to pay for on-chain voting, who might otherwise be unable to express their preferences

## Technical aspects of Autonolas governance module

Autonolas governance module follows the standard governance setup by OpenZeppelin. The following set of contracts are part of the Autonolas Protocol governance module: the veOLAS virtualized-token, the GovernorOLAS, and the Timelock. Here, we will highlight some aspects of this architecture.

- **Non-Upgradable** The GovernorOLAS is derived from the OpenZeppelin modular system of governance contracts. The Governor contract implements the core mechanics of the governance module. It's a non-upgradeable contract. Instead, if it ever needs upgrading, to adapt to future community needs or fixing bugs, the governor can be replaced via a governance vote (using the old governor).
- **Administrative powers** Secure access controls play a significant role in the security of governance modules, usually having sensitive parameters. The contract allows an administrator account to change a number of sensitive parameters: the voting period, the proposal threshold, and the voting delay.  
Note: the administrator account is the Timelock contract, so changes to governance itself are time-delayed.
- **Community multisig** Some changes to the Autonolas Protocol will not go through governance. Instead, they would be executed by a multisig wallet made up of a set of trusted actors. It is expected from the multisig trustees that they limit this power to a set of specific, sensitive changes in the protocol, which need to be applied without being able to rely on the formal governance process. This allows to bypass governance in certain aspects, e.g. security exploits that need to be patched without governance discussion.

The community multisig has the necessary administrative roles to initiate, execute and cancel proposals. This multisig is authorized by the Timelock contract and is subject to the community-defined minimum time delay. After the time delay has passed, the multisig approved proposal can be executed.

Governance-sensitive parameters of the Governor contract itself cannot be changed using the community multisig.

## ARC

Autonolas Request for Comments (ARCs) is a design document providing information to the Autonolas community for proposing and discussing new features or collecting community technical input on an issue. An ARC submission must be announced in the Autonolas governance forum and has the scope of discussion with the community members and receive feedback on the proposal.

We intend an ARC to be the primary step in submitting an Autonolas Improvement Proposal (AIP) that requires an on-chain governance vote to be approved. Optionally, the submission of an ARC can be followed by an off-chain signaling vote to gauge the interest of community participants in the proposal.

### Submission of an ARC

When submitting an ARC on the governance forum the title of the forum post must include 'ARC (**category of the ARC**): title of the ARC ' and the category of the ARC can be chosen from one of the following

- **core** categorizes any change that affects most or all Autonolas implementations, such as a change to the protocol, proposed application standards/conventions, or any change or addition that affects the applications using Autonolas. This further splits in two subcategories
  - **On-chain:** any change that affects the implementation of the on-chain protocol
  - **Off-chain:** any change that affects the implementation of the Autonolas off-chain stack
- **meta** applies to areas other than the Autonolas protocol itself. They may propose a new or missing implementation, but it does not affect the Autonolas codebase. Examples include changes to the decision-making process and broadening of the tools used in Autonolas development
- **informational** describes an Autonolas design issue, or provides general guidelines or information to the Autonolas community, but does not propose a new feature

Note that the post ARC with category **core** should cover all the information of the potentially final AIP.

The ARC must include

- Short and concise title of the ARC
- The creation date of the ARC
- The category of the proposal ( e.g. **core**, **meta**, **informational**)
- A short and concise description of the proposal
- The rationale for the ARC

Optionally the ARC can include

- Technical specifications and optionally implementation details of a new feature
- Security considerations that might be important for security discussions, and risks that are relevant throughout the life cycle of the proposal

## Commenting and discussion period

Once posted on the governance forum, all questions and comments should be replied to and taken into consideration, to improve the ARC further.

A governance off-chain signaling vote can be considered in order to gather community feedback and the preferred path. This allows reaching a rough consensus to maximize the chances of a positive outcome of a possible AIPs submission.

## Recommendations

It is highly recommended that a single ARC contain a single key proposal or new idea. The more focused the ARC the more successful a possible AIP submission tends to be.

## AIP

AIP stands for Autonolas Improvement Proposal. An AIP is a design for proposing new features to the Autonolas protocol and for documenting the design decisions that have gone into Autonolas.

AIPs may be created and submitted by any community member who has sufficient voting power (at least 5000 veOLAS for the moment). It is recommended to start with an ARC

(with the category **core**) to gather feedback on the governance forum and optionally wait for the results of an off-chain signaling vote to reach a rough consensus before finally moving to the AIP.

## Recommendations

As for ACR, it is highly recommended that a single AIP contain a single key proposal or new idea. The more focused the AIP is, the more successful it tends to be.

An AIP must meet certain minimum criteria. It must be a clear and complete description of the proposed enhancement. The enhancement must represent a net improvement. The proposed implementation, if applicable, must be solid and must not add unnecessary complexity to the protocol or the code base.

### Submission of an AIP

For an Autonolas Improvement Proposal (AIP) to be eligible the guide on the Autonolas/AIP github repository must be followed.

The AIP number will be assigned by an AIP reviewer, and the AIP status should start as a WIP (Work In Progress). The AIP should include all the following relevant information

- Short and concise title of the AIC
- The creation date of the AIC
- Links to the forum on the corresponding discussion and eventually to the off-chain signaling vote
- A short and concise description of the proposal
- The motivation for the ARC
- The technical specifications and eventually the implementation details of the new feature
- The rationale for the ARC
- Testing for implementation is mandatory for AIPs but can be included with the implementation
- Implementation of the new feature
- Security considerations that might be important for security discussions, as well as attack vectors

At the end of the doc there is a suggested template for new AIPs.

## Preparation and submission of AIP for on-chain vote

Once the AIP is written and polished, it needs to be submitted via Github by forking the AIP repository and following the instruction there which, in particular, require creating an AIP pull request.

Once the AIP pull request has been polished and reviewed, it becomes ready for an on-chain governance vote. A community member with enough voting power is now able to submit the AIP to be voted on. This is a technical process, so if you're not sure about the technical details of submitting your proposal to the Autonolas DAO, you can ask in the Autonolas Discord for help from the community.

## Document History

The second part of this document was inspired and partly derived from the AAVE governance documentation ( <https://docs.aave.com/governance/arcs>, <https://docs.aave.com/governance/aips> ) and also inspired by the "EIP Purpose and Guidelines" for the *Ethereum Improvement Proposals* (<https://eips.ethereum.org/EIPS/eip-1#what-is-an-eip>)