

# Cross-chain governance: from Ethereum to Polygon

This document summarizes how Autonolas will handle cross-chain governance between Ethereum to Polygon.

## Background

Currently, all ownable contracts belonging to the on-chain Autonolas protocol and deployed on Ethereum, are under the ownership of the [Autonolas Timelock contract](#). Therefore, a successful on-chain governance proposal is sufficient to take action on such contracts.

Eventually, some contracts will be deployed by the Autonolas DAO on other smart contract chains, e.g., a lightweight part of the Autonolas Registry will be deployed on the Polygon mainnet in order to enable registration of autonomous service NFTs on the Polygon mainnet as well as management of services on-chain. In such events, it is necessary to allow the Autonolas DAO to control such Autonolas contracts deployed on Polygon.

In order to do this, Autonolas will use the [FxPortal](#) developed and designed by the Polygon team to support cross-chain bridging from Ethereum to Polygon.

## FxPortal

As described in [Polygon wiki: FxPortal Overview](#), the FxPortal is a simple implementation of the Polygon [state sync](#) mechanism that allows any state-syncs without mapping. FxChild and FxRoot (cf. [fx-portal/contracts](#) for the contracts and the addresses on mainnets & testnets) are the main contracts on which mapping-less bridge works. FxPortal calls and passes data to user-defined methods on another chain without mapping.

## Autonolas Cross-Chain governance

Autonolas will deploy a contract to Polygon, *FxGovernorTunnel*, that will be able to execute on such a chain the successful governance proposal on Ethereum mainnet. In summary, to act on Autonolas contracts deployed on Polygon, the following steps are necessary:

1. have a governance proposal on Ethereum mainnet
2. once the successful proposal is executed on Ethereum, this will be broadcasted by the Timelock to the Autonolas executor contract, *FxGovernorTunnel*, deployed on Polygon mainnet

3. *FxGovernorTunnel* will execute the accepted action on Polygon being the owner of all the contracts deployed on Polygon.

#### Technical workflow

1. A governance proposal will be made on the [GovernorOLAS](#) on Ethereum. The proposal is an encoded call to the method *sendMessageToChild* of the Polygon [FxRoot](#) contract deployed on Ethereum. The calldata of the proposal contains two encoded variables:
  - a. The first variable is the address of the Autonolas executor contract on Polygon, i.e. *FxGovernorTunnel*, that will decode and process the message on the Polygon chain.
  - b. The second variable is the data that will be decoded on the Polygon chain. This field contains encoded bytes for the following contiguous array of fields: *[target, value, payloadLength, payload]* (see [Test for FxGovernorTunnel](#) and [Test fx\\_goerli\\_mumbai\\_governor.js](#) for test-case examples)
2. When a successful proposal is executed, the Autonolas timelock triggers the *sendMessageToChild* call on the Polygon [FxRoot](#) contract, which, in turn, triggers *syncState()* on the Polygon [StateSender](#) contract deployed on Ethereum. This will trigger the emission of a *StateSync* event.
3. Polygon validators listening for this *StateSync* event then trigger the *onStateReceived()* in the Polygon [FxChild](#) contract deployed to Polygon.
4. In *onStateReceived()* the encoded data (produced in part 1) is passed to *FxGovernorTunnel*, the Autonolas executor contract deployed on Polygon, that implements the function *processMessageFromRoot* to process the message received.

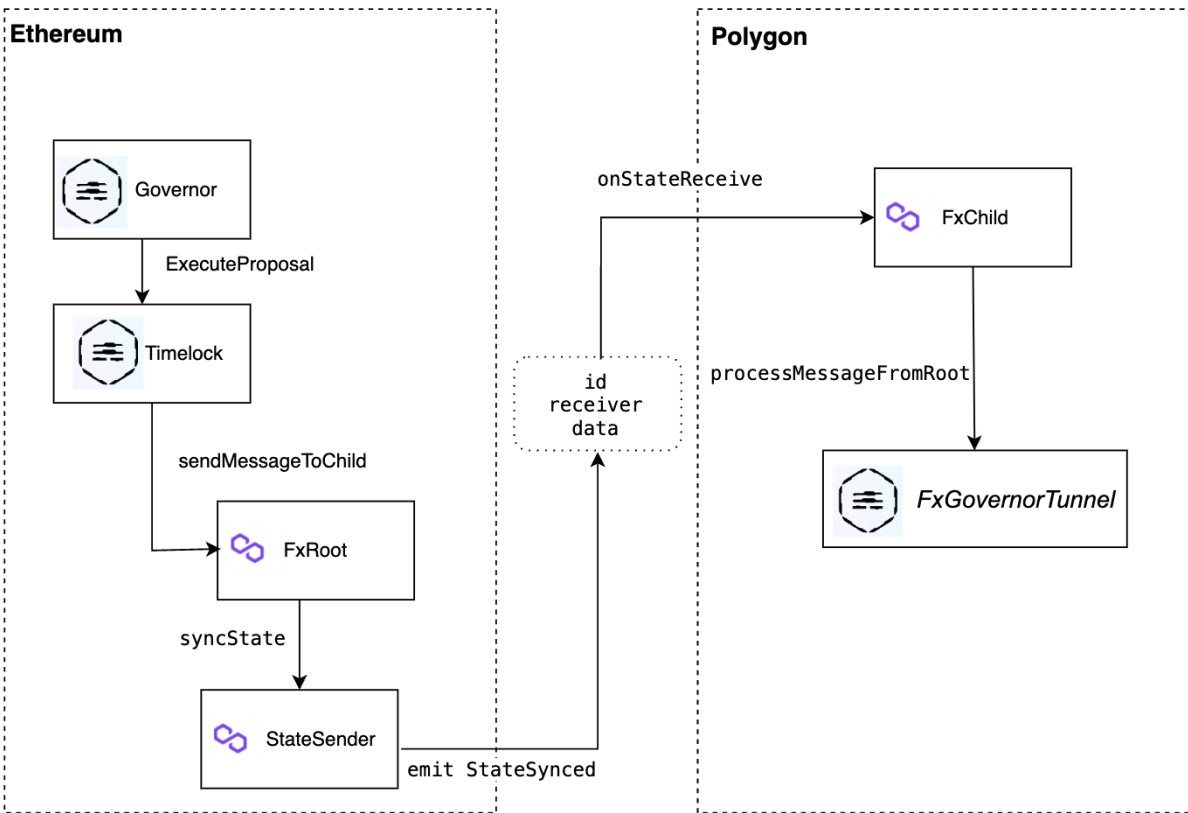


Fig. Autonolas cross-chain governance architecture from Ethereum to Polygon

#### References

1. [FxPortal Overview](#)
2. [Uniswap owner on Polygon: EthereumProxy contract](#)
3. [Uniswap L1 owner: Timelock on Ethereum](#)
4. [OZ: adding cross-chain support](#)
5. [Aave:governance-crosschain-bridges](#)