Cross-chain governance

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 different L2 chains. For example, a lightweight part of the Autonolas Registry is being deployed on the Polygon and Gnosis mainnets in order to enable registration of autonomous service NFTs, as well as their management of services on-chain. In such events, it is necessary to allow the Autonolas DAO to control L2 Autonolas contracts.

This document describes the means of cross-chain connection between the Autonolas governance on L1 and corresponding L2 protocol contracts owned by the L1-L2 mediator contracts that process messages specifically from the main L1 Timelock contract.

Ethereum to Polygon

In order to handle the cross-chain governance between Ethereum and Polygon networks, Autonolas will use the FxPortal developed and designed by the Polygon team to support cross-chain bridging between Ethereum and Polygon.

FxPortal

As described in Polygon wiki: FxPortal Overview, the FxPortat 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 on Polygon

Autonolas will deploy a contract on 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 step 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 s*yncState()* 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.

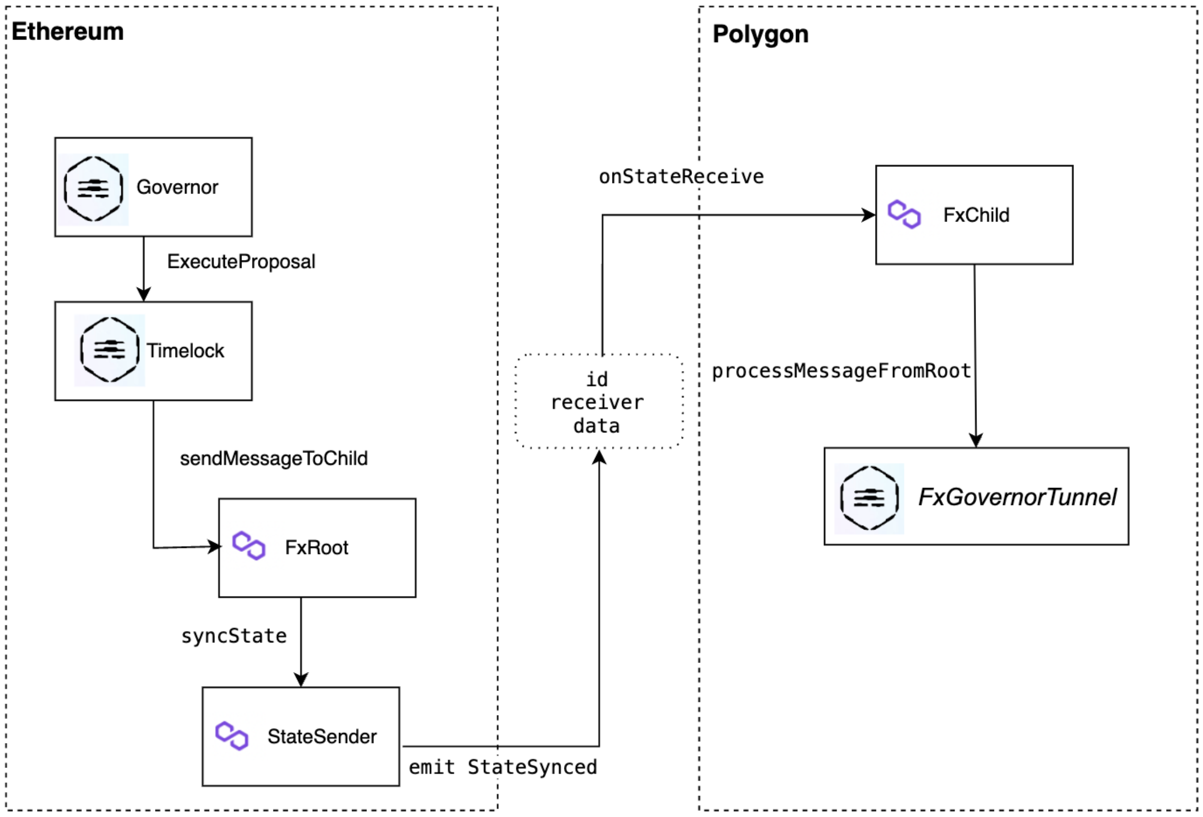

Fig. Autonolas cross-chain governance architecture from Ethereum to Polygon

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

Ethereum to Gnosis Chain

In order to handle the cross-chain governance between Ethereum and Gnosis networks, Autonolas will use the Arbitrary Message Bridge (AMB) technique developed and designed by the Gnosis Chain team to support cross-chain bridging between Ethereum and Gnosis.

AMB Bridge

AMB Bridge is a set of native Ethereum and Gnosis Chain proxy contracts with the implementation called *EternalStorageProxy,* cf. EternalStorageProxy (Home) on Gnosis network and EternalStorageProxy (Foreign) on Ethereum network The chain connecting contracts can be observed in the corresponding section of the documentation. To have a cross-chain governance between Ethereum and Gnosis chain, the idea is to deploy a contract on the L2 chain that receives and processes messages from the L1, and executes them on behalf of the Timelock contract.

Autonolas Cross-Chain governance on Gnosis

Autonolas will deploy a contract on Gnosis, *HomeMediator*, 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 Gnosis, the following step 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, *HomeMediator*, deployed on the Gnosis mainnet
3. *HomeMediator* will execute the accepted action on Gnosis being the owner of all the contracts deployed there.

Technical workflow

1. A governance proposal will be made on the GovernorOLAS on Ethereum. The proposal is an encoded set of calls with the method *processMessageFromForeign()*, additionally encoded with the *requireToPassMessage()* method with the target on the AMB Contract Proxy (Foreign) deployed on Ethereum. The calldata of the proposal contains three encoded variables:
    a. The first variable is the address of the Autonolas executor contract on Gnosis, i.e. *HomeMediator*, that will decode and process the message on the Gnosis chain.

   b. The second variable is the *processMessageFromForeign()* encoded data that will be decoded on the Gnosis chain via the *HomeMediator* contract. The data field contains encoded bytes for the following contiguous array of fields: *[target, value, payloadLength, payload]* (see Test for HomeMediator and Test mediator_goerli_chiado_governor.js for test-case examples).
   c. The third parameter is the amount of gas to be provided in execution of the method call on the other side.

2. When a successful proposal is executed, the Autonolas' Timelock contract triggers the *requireToPassMessage()* call on the Ethereum AMB Contract Proxy (Foreign) contract, which, in turn, triggers *UserRequestForAffirmation* event on Ethereum.
3. Gnosis validators ensure the message receival between Ethereum and Gnosis corresponding AMB Contract Proxies. Specifically, *executeAffirmation()* on the Gnosis AMB Contract Proxy (Home) with the *processMessageFromForeign() encoded* data will be triggered.
4. Finally, the Autonolas' *HomeMediator* executor contract deployed on Gnosis implements the function *processMessageFromForeign()* which allows to process the message received.

References

5. Gnosis Chain Docs
6. Arbitrary Message Bridge