Bonding mechanism with Polygon LP-token: bridging LP-token created on Polygon to Ethereum and vice-versa

Motivation

Autonolas protocol is already deployed on various blockchains including Ethereum, Gnosis, Polygon, and Solana.

Autonolas employs cross-chain governance to enable governance actions between Ethereum (L1) and L2 networks like Polygon and Gnosis Chain at present, and Solana in the future. For cross-chain governance, Autonolas sends messages from Ethereum-based governance contracts to the target chains, employing mechanisms like the FxPortal for Polygon and Arbitrary Message Bridge (AMB) for Gnosis Chain. Further details on Autonolas cross-chain bridging design can be found here.

According to aip-1, here the workflow to enable the bonding of pairs on different chains:

1. Move OLAS token from Ethereum to the target chain, using bridges with low trust requirements, good security, and minimal manual and team interaction to start the process.
2. Create an LP-token on the target chain with the bridged OLAS token on the target chain and use popular decentralized exchanges with a Uniswap-v2-style AMM design.
3. Transfer the LP-token back to Ethereum using the same bridge methods.
4. Use the transferred LP-token for bonding in Autonolas' mechanism on Ethereum.

In order to apply this workflow between Ethereum and Polygon

1. OLAS token on Ethereum can be bridged to Polygon by using native bridge solution developed from Polygon team (cf. fx-portal/contracts and ERC20 token tranfer from Ethereum to Polygon).
2. While, to transfer an LP token deployed on Polygon to Ethereum based on a Polygon-native FxPortal, the following steps have been necessary.

a. Revisit FxERC20ChildTunnel.sol and FxERC20RootTunne.sol original contracts implementing
    i. FxERC20ChildTunnel.sol based on the original FxBaseChildTunnel.sol, and
    ii. FxERC20RootTunnel.sol based on the original FxBaseRootTunnel.sol.
b. Implement BridgedERC20.sol token contract that serves as a mapping of Polygon LP token on L1.

An overview of the design and the relevant contract methods are in the following section.

Design overview

FxERC20RootTunnel.sol is going to be the owner of the BridgedERC20.sol contract and acts as a minter of mapped LP tokens on Ethereum.

- On the Polygon side, the LP tokens are going to be locked in the FxERC20ChildTunnel.sol contract, while their mapped tokens are going to be minted in BridgedERC20.sol on the Ethereum network.
- On the Ethereum side, when bringing the tokens representing the LP back to the Polygon, the corresponding token amount is going to be burnt in BridgedERC20.sol on the Ethereum network, and unlocked and transferred on the Polygon side.

The described cross-chain interaction is depicted below.

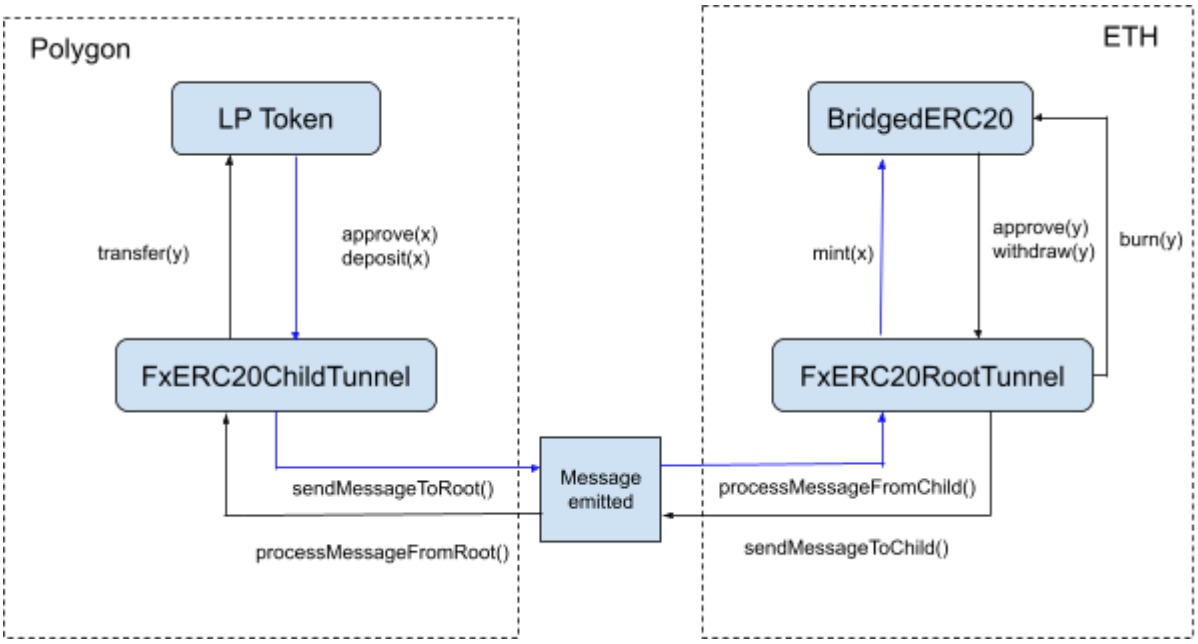

Fig 2. Polygon-Ethereum token transfer interactions.

Polygon to Ethereum deposit()

Here is the interaction that is needed to bridge from the Polygon side:

Polygon side

1. On-chain: Approve x LP tokens for the FxERC20ChildTunnel contract;
2. On-chain: Call FxERC20ChildTunnel.deposit(x);
3. Off-chain: Get the hash of the tx from step 2 (txHash);
4. Off-chain: From the tx from step 3 get the topic 0 hash of the MessageSent() event (msgHash);
5. Off-chain: using txHash and msgHash, monitor the call to the polygon API for the tx proof to become valid (might take 30 minute or more): https://proof-generator.polygon.technology/api/v1/mumbai/exit-payload/0xe93b341a8ae0ecb7b1c90dd1df022e5f9fcfc851074a277229700afaf5c09b0b?eventSignature=0×8c5261668696ce22758910d05bab8f186d6eb247ceac2af2e82c7dc17669b036
6. Off-chain: record the result value of step 5 as inputData.

Ethereum side

1. On-chain: call FxERC20RootTunnel.receiveMessage(inputData);
2. On-chain: x BridgesERC20 tokens are minted to the receiver on Ethereum side.

ETH to Polygon withdraw()

Here is the interaction that is needed to bridge back from the Ethereum side:

Ethereum side

1. On-chain: Approve y bridged LP tokens for the FxERC20RootTunnel contract;
2. On-chain: Call FxERC20RootTunnel.withdraw(y). Then y tokens are burnt and the token transfer message is sent to the Polygon bridge.

Polygon side

1. On-chain: Polygon StateSync does everything automatically. The y tokens will be transferred to a receiver.

Facilitating the bridging process

This repository https://github.com/valory-xyz/erc20-token-bridge contains the instruction, the script, and the configuration for the networks to facilitate in bridging tokens from Polygon to Ethereum and vice versa.