

Smart Contract Outline

1. Initialize Libraries and Imports

- Import OpenZeppelin contracts for ERC20, AccessControl, and other utility libraries.
- Integrate Pyth Network oracle price feeds for real-time data.

2. Define Contract Variables

- Set token properties (name, symbol, etc.).
- Define the oracle address and data struct to hold price information.
- Thresholds for triggering minting or burning actions based on deviation from the peg.

3. Access Control Setup

- Use roles (such as ADMIN_ROLE and OPERATOR_ROLE) to manage access to minting and burning functions.

4. Constructor and Initialization Functions

- Initialize the contract with a specific peg price and price tolerance.
- Assign initial roles for minting and burning.

5. Mint and Burn Functions

- Implement mint and burn functions, restricted to approved roles.
- Define a rebalance function that checks the price deviation and calls mint or burn as needed.

6. Oracle Price Check Function

- Fetch the current price from the Pyth Network.
- Compare it to the peg, determining if minting or burning should occur.

7. Rebalancing Logic

- Set price deviation thresholds to trigger minting or burning.
- Implement adjustable parameters to handle different levels of market volatility.