****EVM Token Bridge with FE****

## Technological Level with Smart Contracts (Hardhat project)

### Minimum 2 contracts:

* A contract for Bridge itself -> BridgeFactory.sol
* A contract for deploying a wrapped ERC20 token -> WERC20.sol
* Unit tests for both contracts with > 95% code coverage
* Events need to be emitted on each major action into contracts

### Deployment Scripts:

* Automatic deployment scripts for desired testnets

## Technological level with dApp

### SDK (Optional):

* contract interactions for the main functions of the contract
* view functions, where you’ve got additional formatting of the data (for example checks, BN conversions, and metadata parsing)

### Application Level:

* As a user, I want to select a token (from my wallet or by address):
  + There should be a separate “Transfer” view (can be a page, tab or other)
  + List ERC-20 tokens (names) from the wallet (in a dropdown for example)
  + The selected ERC-20 contract address is shown in a text field
  + Give the option to manually paste a token contract address
* As a user, I can select a target chain:
  + List target chains in a dropdown
  + The current active chain should be grayed out in the dropdown
* As a user, I want to be able to lock tokens in the bridge contract:
  + Should happen in “Transfer” view
  + When the Send button is clicked, three things should happen:
    - Approve the bridge contract as a token spender for the user’s wallet
    - Send a transaction that initiates the bridge operation
    - Emit blockchain event about the operation
* As a user, I can connect to the target chain:
  + Provide a button that will initiate the change of the network
* As a user, I want to be able to claim the locked tokens:
  + There should be a separate “Claim” view (can be a page, tab or other)
  + The “Claim” button should be active only if there are tokens to be claimed
  + The button action should match the type of the operation
* As a user, I can bridge the tokens (previously claimed) back to their native chain:
  + Should happen in “Transfer” view
  + The “Send” button should detect if this is a previously claimed token or a native one
  + When the Send button is clicked, three things should happen:
    - Approve the bridge contract as a token spender for the user’s wallet
    - Send a transaction that initiates the bridge operation
    - Emit blockchain event about the operation
* As a user, I can claim the tokens on the native chain and receive the original ones:
  + Should happen in the “Claim” view
  + The “Claim” button should be active only if there are tokens to be claimed
  + The button action should match the type of the operation

# Bonus features

* The solution should be implemented with a stand-alone transaction validator solution.
* As a user, I can see the history of my transactions.
* Use Permits to avoid sending a separate approve transaction

# Completeness of the project:

* Two repositories with FE and Contract Implementation
* Contracts deployed on the desired testnet
* FE connected with those contracts
* Short video with the functionalities.