The challenge will consist of solving the following problem:

Interact with a public **testing MultiversX blockchain** in such a way that the following requirements are met:

1. Create 2 private/public keys (will call them key1 & key2)
2. Request 30 xEGLD tokens from the faucet using the web wallet for both addresses (https://testnet-wallet.multiversx.com/faucet)
3. Create an NFT token from key1
4. Send the NFT token from key1 to key2
5. Send back the NFT from key2 to key1

Although we can solve this problem strictly using the web wallet (https://testnet-wallet.multiversx.com/unlock) without writing a single line of code, points 3-5 should be done by executing a program written in Go.

Technical specs for the Golang app:

- The binary should be able to accept parameters like the gateway URL (example: https://testnet-gateway.multiversx.com), and the key1 & key2 private key files (either the .json & the password or the .pem file). The parameters can be defined as binary flags or .toml/.yml files or both;
- Execute the required transactions and keep track of the results. Signal the errors if something goes wrong and issue a non-0 exit code for the binary;
- All steps should write local logs (either by sending them to stdout or in a file or both). For this step, we already have a library for this https://github.com/multiversx/mx-chain-logger-go you might consider using it or writing your own implementation;
- Hardcode as few parameters as you can (such as the gas limit for the transaction, gas price, chain ID, and so on). Instead, query them on the gateway;
- Make the binary print the transaction hashes so we can easily track them;
- Use GitHub to host the source code and ensure that all the commits are verified.

Optional (desired) considerations:

- Apply as much as you can the clean code/clean architecture principles (some good reads can be found here: http://cleancoder.com/products)
- Add unit tests and/or integration tests

Useful links:

- https://docs.multiversx.com/sdk-and-tools/sdk-go (SDK written in Go that has a few components useful for this task along with some interaction examples) - **not mandatory to use this SDK**
- https://testnet-gateway.multiversx.com/ for exploring the implemented REST API endpoint routes
- https://docs.multiversx.com/tokens/nft-tokens (all about NFTs & SFTs)