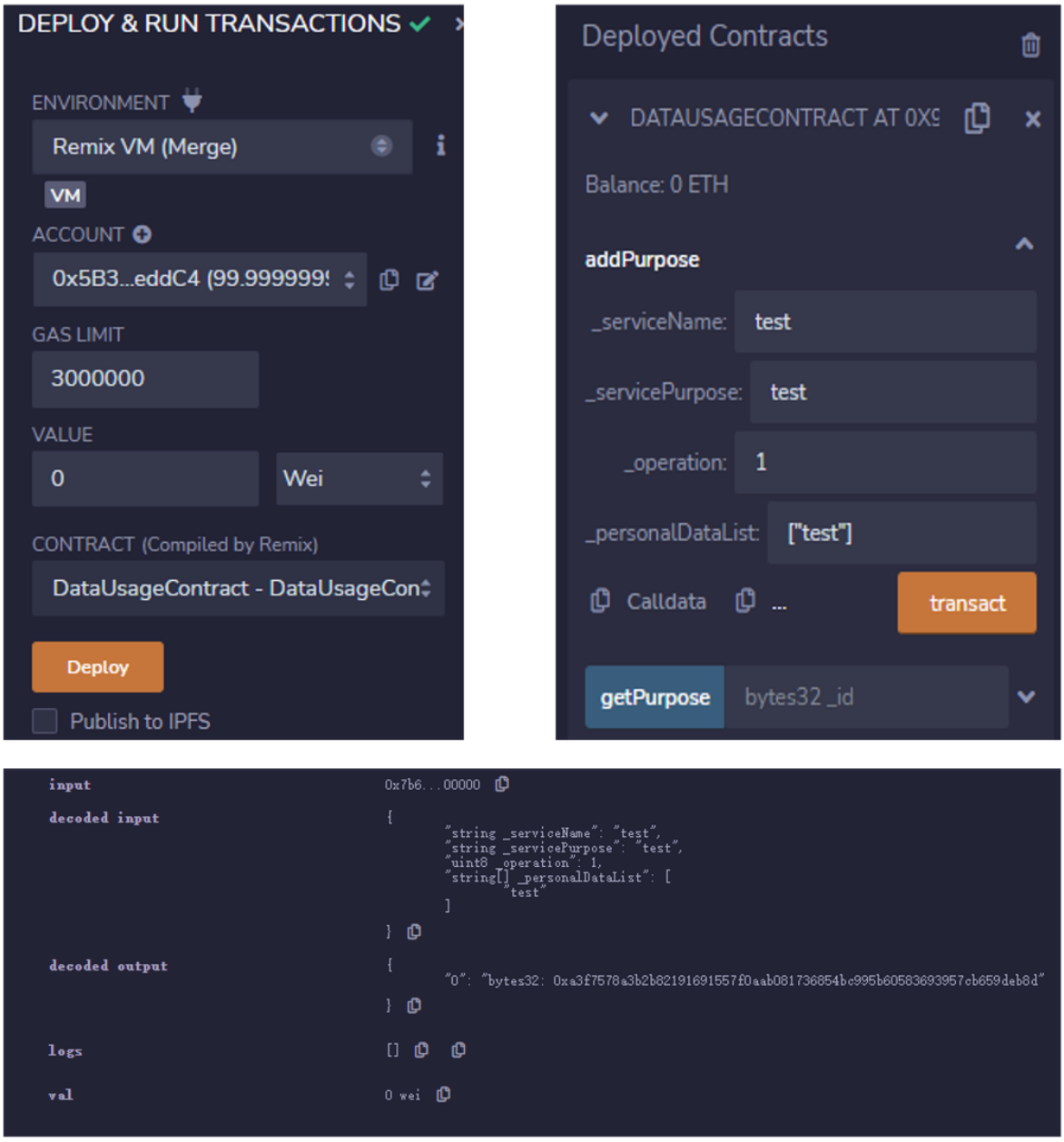
# Deployment Plan

## 1. Deployment process

This report will cover the deployment process in two separate sections, Basic Specification and Extension. In the Basic Specification deployment process, the order in which the three basic contracts are deployed and the considerations are explained. In the Extension deployment process, the focus is on how the Verification Contract is deployed based on the Basic Specification and how the complete deployment process is deployed to the Ethernet public blockchain test network.

### 1.1 Basic Specification

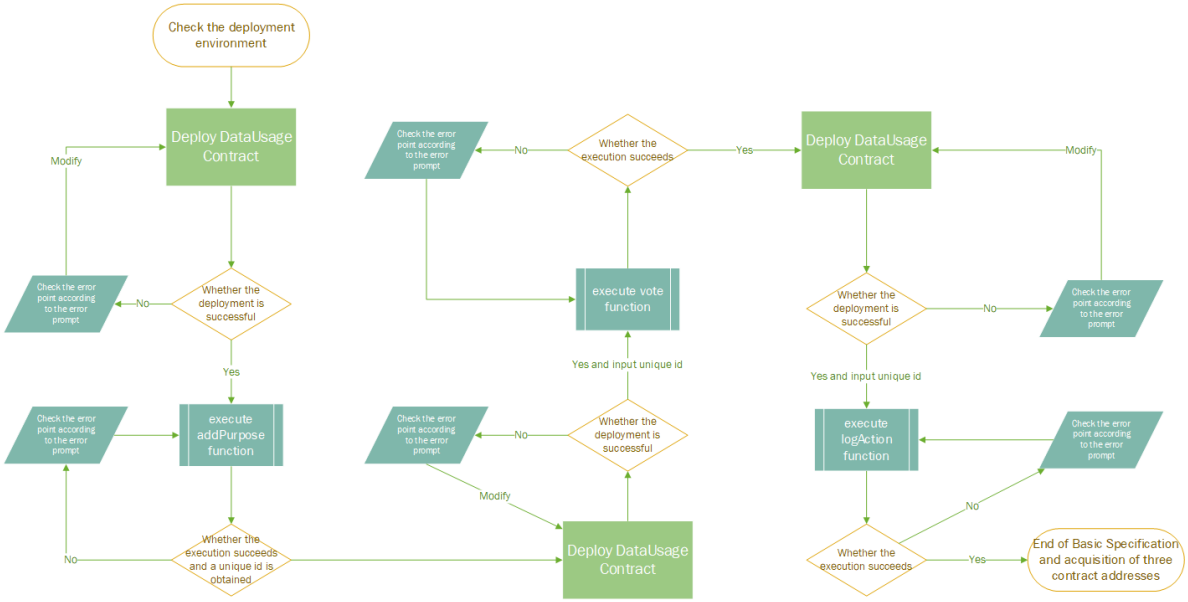
As the Basic Specification is deployed on the Remix IDE, you need to ensure that you have an available Remix VM environment and that you have enough Ether in the virtual account and that all the contracts are compiled and error free before deploying. Once the deployment environment is in order, you will start deploying the 3 smart contracts, starting with the DataUsage Contract. As mentioned earlier, a field 'id' is designed in the DataUsage Contract to ensure the uniqueness of the contract. In an objective transaction process, the user data is first requested according to the purpose of the transaction. As shown in Figure 1, the Data Usage Contract will be successfully deployed when the Deploy button is clicked, and the hashed unique id will be available in the console by filling in the appropriate fields for the addPurpose function. subsequent deployment of the contract will not be illustrated in the picture.




Firgure 1. Deploy DataUsage Contract

Once the unique id is obtained, the deployment of the Agreement Contract and Log Contract will be performed in turn. The deployment process is the same as above, but it is worth noting that the above the unique id need to be passed in when executing the vote function in the Agreement Contract and the logAction function in the Log Contract to ensure the uniqueness of the action. The deployment of the Basic Specification is now complete and the 3 contract addresses have been obtained for the deployment of the first extension (Verification Contract).

The complete deployment flow chart is shown in Figure 2.

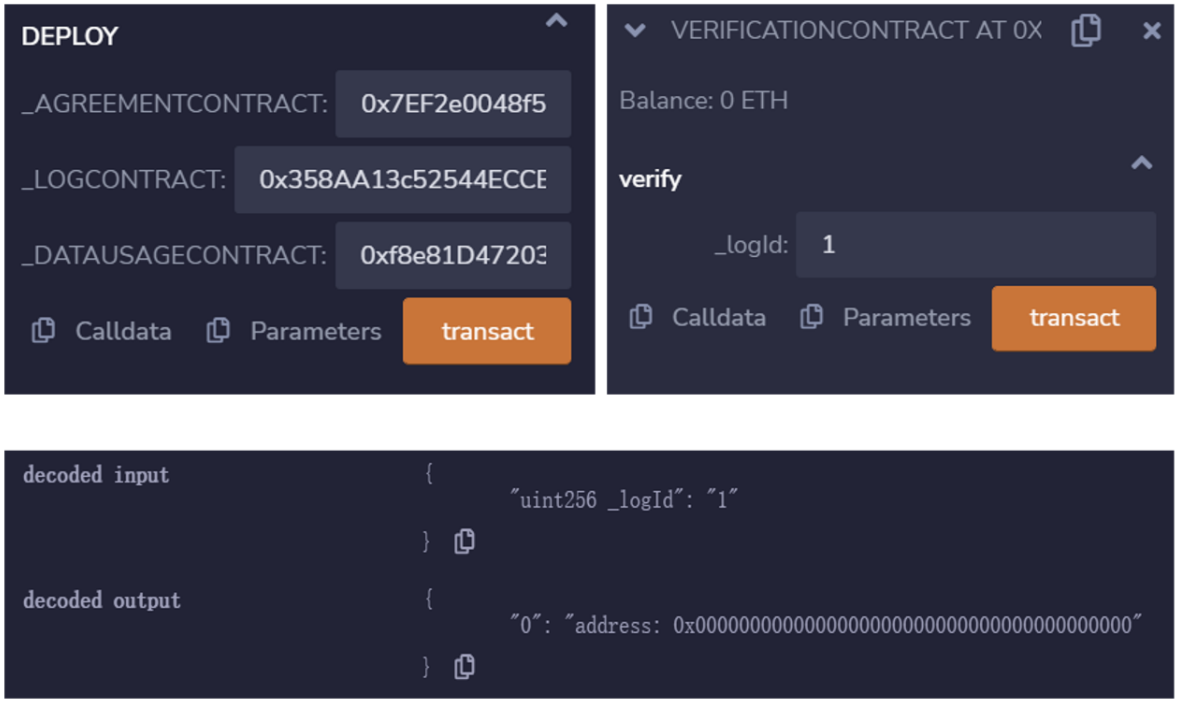



Firgure 2. Basic Specification Flowchart

# 1.2 Extension

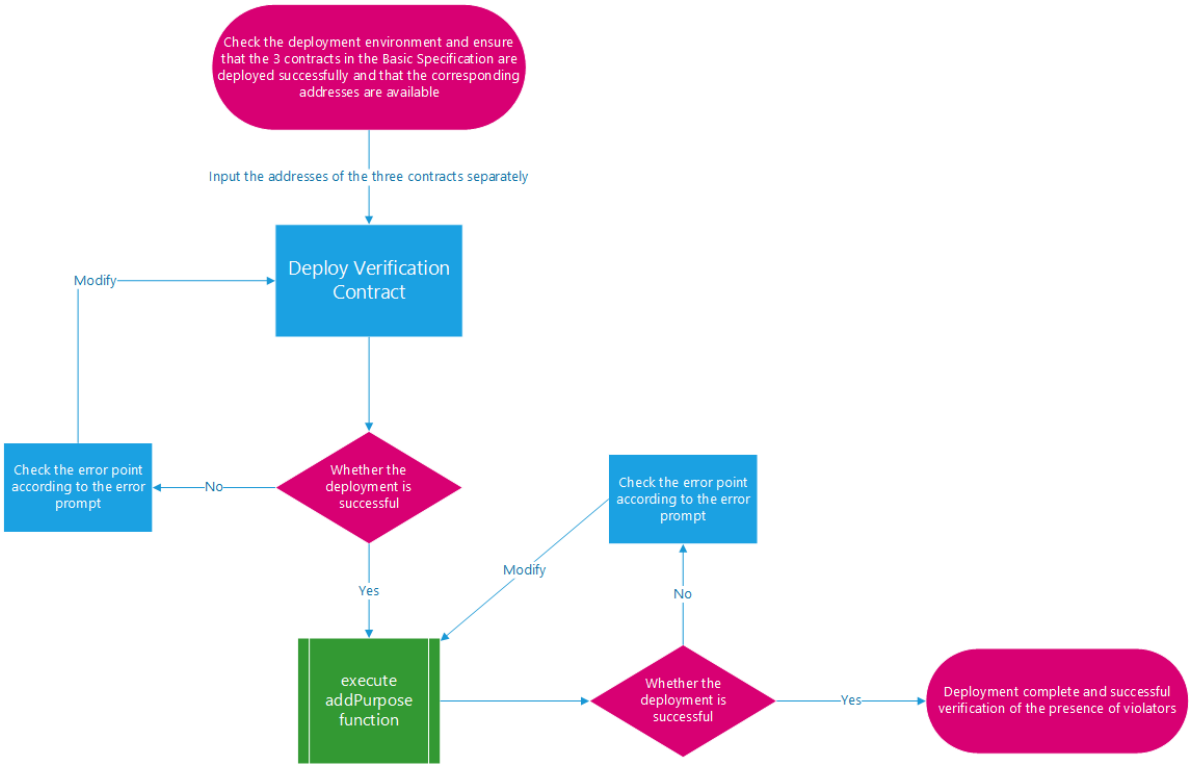
## 1.2.1 First extension (Compliance Verification)

This contract is consistent with the Basic Specification and is all deployed in the Remix IDE. After deploying the Basic Specification, you will be provided with the addresses of three separate contracts. Before deploying the Verification Contract, the three addresses need to be filled in the corresponding boxes in order to successfully create the contract and detect the presence of offenders. This is shown in Figure 3.




Firgure 3. Deploy Verification Contract

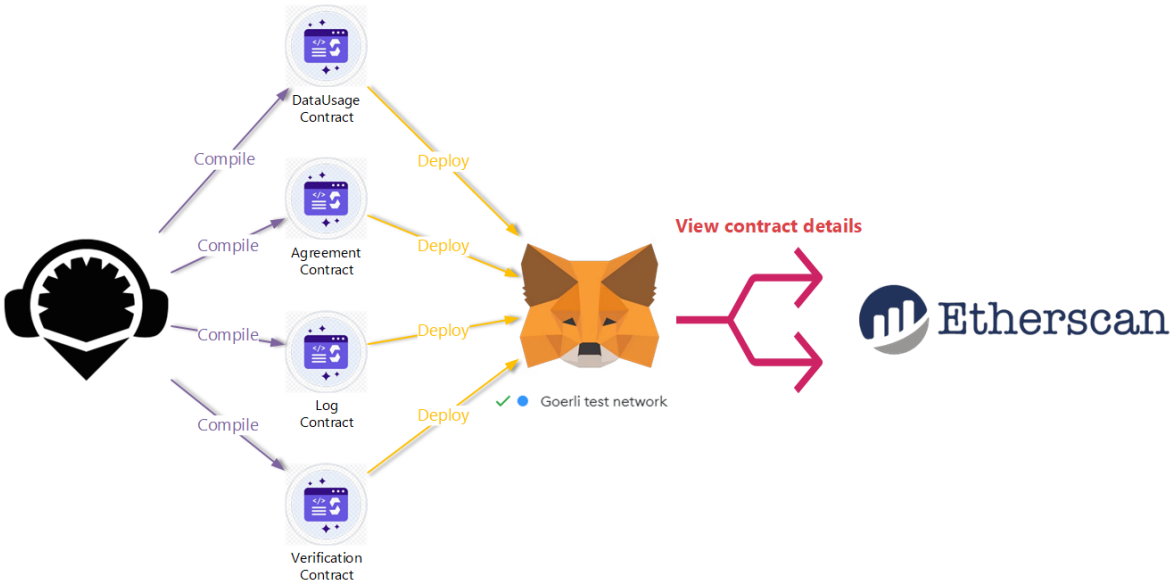
The flow chart is shown in Figure 4.




Firgure 4. First Extension Flowchart

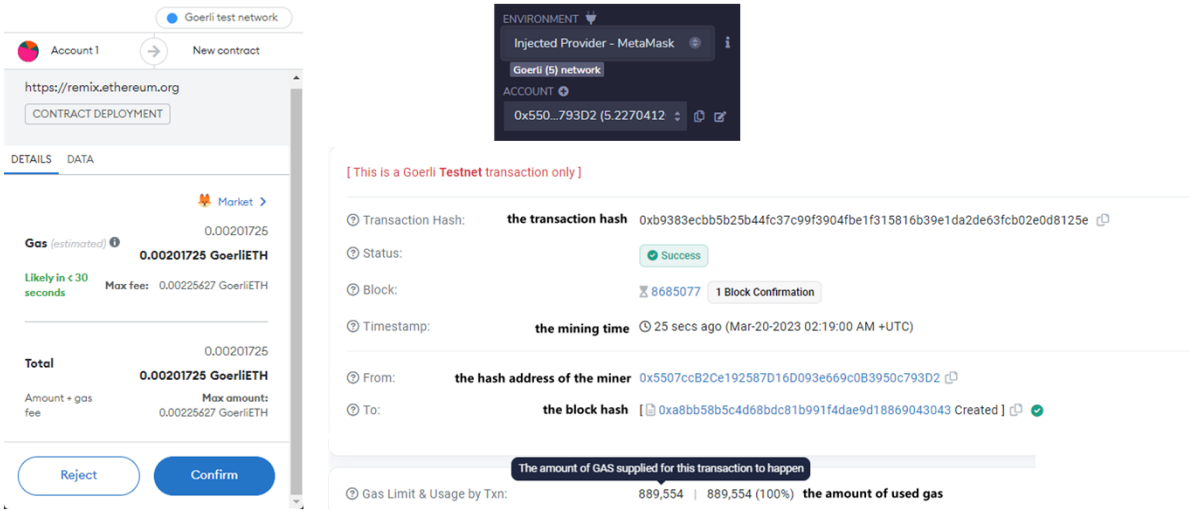
### 1.2.2 Second extension (Making the public Blockchain)

For the second extension, the project uses Remix to write four contracts and connects them to the Goerli public blockchain test network for deployment via MetaMask. The architecture diagram is shown in Figure 5.



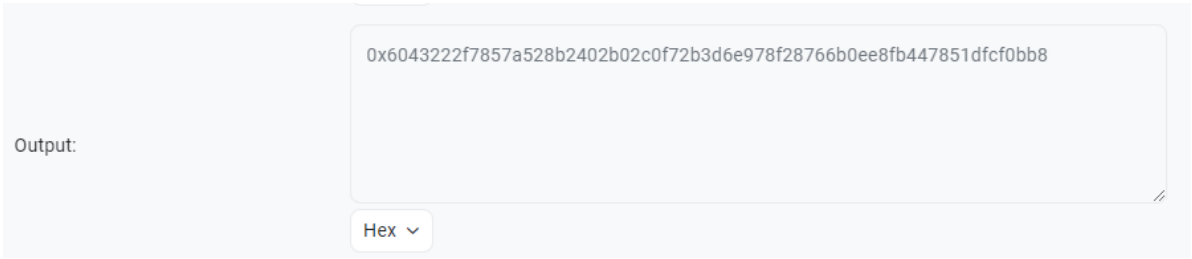

Firgure 5. Public Blockchain structure

Before starting the deployment, you need to select the test network as 'Injected Provider' in the Remix IDE to be able to call the test network in MetaMask, click on the Deploy button and the deployment details will pop up in the MetaMask client. Click on the Comfirm button to start the deployment. The exact deployment process can be viewed in Etherscan. At the end of the deployment, the following information can be viewed in Etherscan (i) the block hash, (ii) the transaction hash, (iii) the hash address of the miner, (iv) the amount of used gas, and (v) the mining time. As shown in Figure 6. Subsequent deployment steps will not be shown in screenshots.




Firgure 6. Deploy on public blockchain

The whole deployment process is the same as described in Basic Specification and First Extension above. It is important to note that at this point, after executing the function in the Remix IDE, the return value will not be displayed as an output in the Remix IDE terminal, this output should be viewed in Etherscan. For example, check the unique id returned after executing the addPurpose function in the DataUsage Contract, which is shown in Figure 7.




Firgure 7. Check output on Etherscan