

High severity issues

- **Initial Token Distribution / Centralization**

In the constructor function, the deployer is minting tokens for tokenOwnerAddress. All the total supply are sent to the tokenOwnerAddress when deploying the contract. This could be a centralization risk, the tokenOwnerAddress can distribute those tokens without obtaining the consensus of the community.

```
1      constructor(  
2          string memory name,  
3          string memory symbol,  
4          uint8 decimals,  
5          uint256 totalSupply,  
6          address payable feeReceiver,  
7          address tokenOwnerAddress  
8      ) public payable {  
9          _name = name;  
10         _symbol = symbol;  
11         _decimals = decimals;  
12  
13         // set tokenOwnerAddress as owner of all tokens *****  
ISK  
14         _mint(tokenOwnerAddress, totalSupply);  
15  
16         // pay the service fee for contract deployment  
17         feeReceiver.transfer(msg.value);  
18     }
```

- ✓ **Recommendation**

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key, We also advise the client to adopt Multisig, Timelock, and/or DAO in the project to manage the specific account in this case.