CLOUET
Baudouin

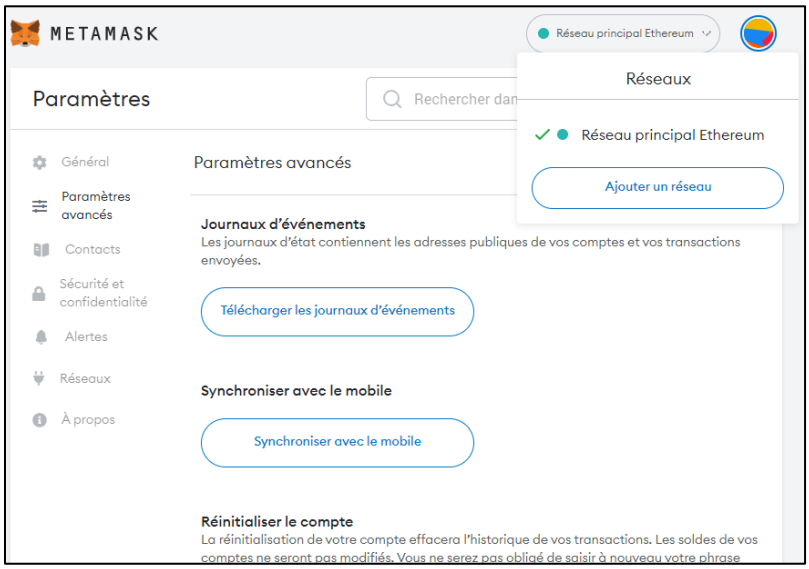
TP Blockchain :

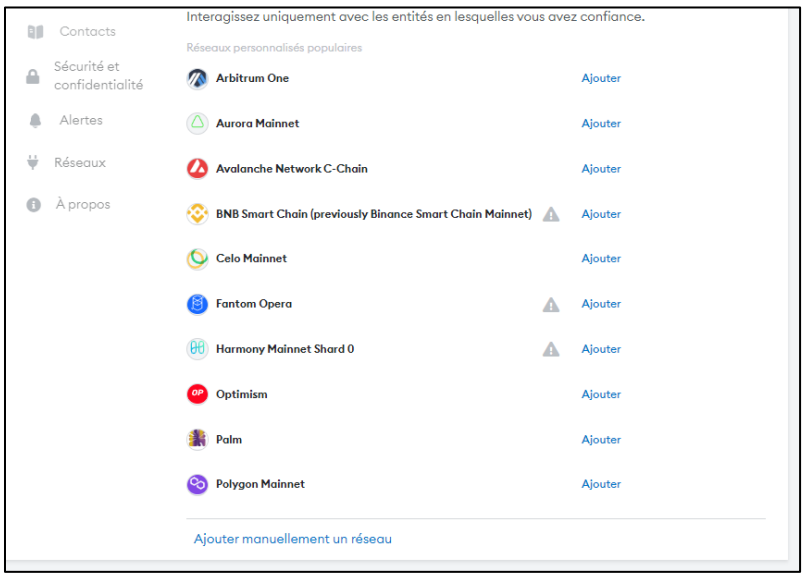
Développer, Deloyer et Interagir avec un contrat intelligent sur Ethereum

d)

Clé publique : 0xca89932fD0758A0A8BB88F1CdE8237B1105Bf341

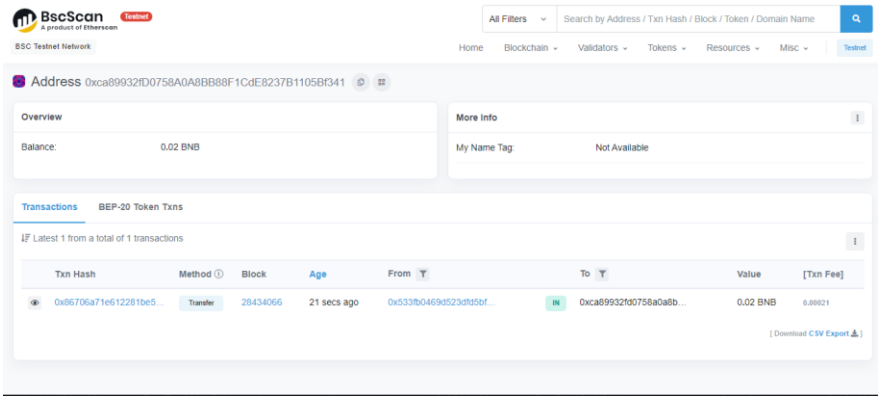
- <u>Ajout du nouveau réseau :</u>





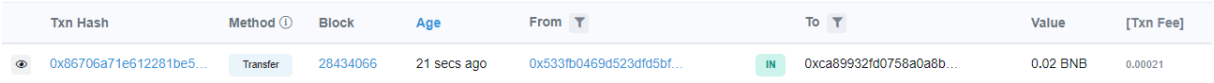
f)

- <u>Transaction générée sur le compte :</u>



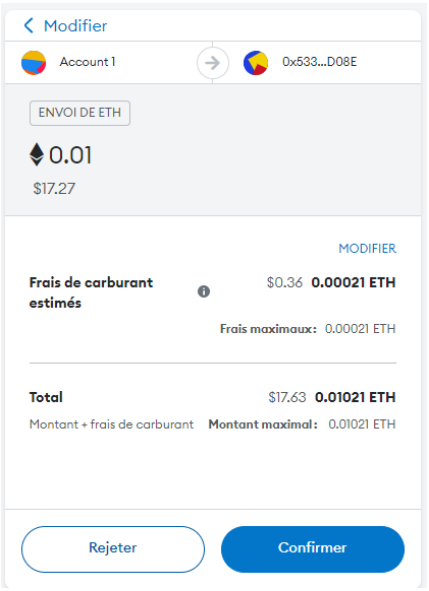
g)

- <u>Numéro de bloc :</u>



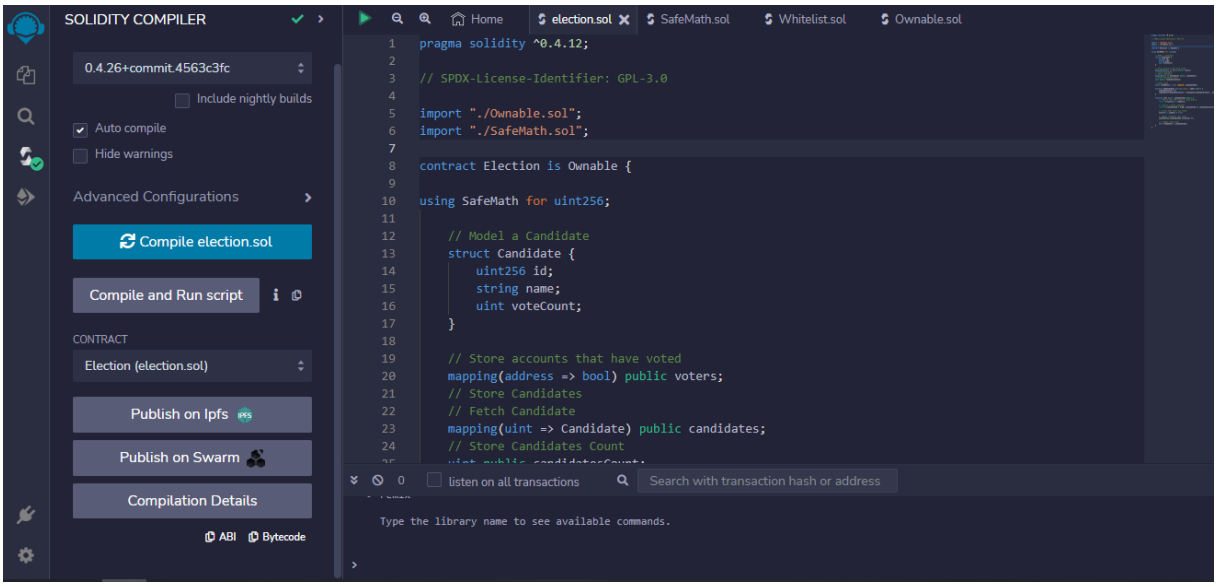
h)

- <u>Première transaction :</u>



I)

- <u>Compilation du smart contract :</u>



- <u>ABI :</u>


```
[
	{
		"constant": false,
		"inputs": [
			{
				"name": "_candidateId",
				"type": "uint256"
			}
		],
		"name": "vote",
		"outputs": [],
		"payable": false,
		"stateMutability": "nonpayable",
		"type": "function"
	},
	{
		"constant": true,
		"inputs": [],
		"name": "candidatesCount",
		"outputs": [
			{
				"name": "",
```

```
                        "type": "uint256"
                }
            ],
            "payable": false,
            "stateMutability": "view",
            "type": "function"
        },
        {
            "constant": true,
            "inputs": [
                {
                    "name": "",
                    "type": "uint256"
                }
            ],
            "name": "candidates",
            "outputs": [
                {
                    "name": "id",
                    "type": "uint256"
                },
                {
                    "name": "name",
                    "type": "string"
                },
                {
                    "name": "voteCount",
                    "type": "uint256"
                }
            ],
            "payable": false,
            "stateMutability": "view",
            "type": "function"
        },
        {
            "constant": false,
            "inputs": [
                {
                    "name": "_name",
                    "type": "string"
                }
            ],
            "name": "addCandidate",
            "outputs": [],
            "payable": false,
            "stateMutability": "nonpayable",
```

```
		"type": "function"
	},
	{
		"constant": true,
		"inputs": [],
		"name": "owner",
		"outputs": [
			{
				"name": "",
				"type": "address"
			}
		],
		"payable": false,
		"stateMutability": "view",
		"type": "function"
	},
	{
		"constant": true,
		"inputs": [
			{
				"name": "",
				"type": "address"
			}
		],
		"name": "voters",
		"outputs": [
			{
				"name": "",
				"type": "bool"
			}
		],
		"payable": false,
		"stateMutability": "view",
		"type": "function"
	},
	{
		"constant": false,
		"inputs": [
			{
				"name": "newOwner",
				"type": "address"
			}
		],
		"name": "transferOwnership",
		"outputs": [],
		"payable": false,
```

```
                "stateMutability": "nonpayable",
                "type": "function"
        },
        {
                "anonymous": false,
                "inputs": [
                        {
                                "indexed": true,
                                "name": "_candidateId",
                                "type": "uint256"
                        }
                ],
                "name": "votedEvent",
                "type": "event"
        },
        {
                "anonymous": false,
                "inputs": [
                        {
                                "indexed": true,
                                "name": "previousOwner",
                                "type": "address"
                        },
                        {
                                "indexed": true,
                                "name": "newOwner",
                                "type": "address"
                        }
                ],
                "name": "OwnershipTransferred",
                "type": "event"
        }
]
```

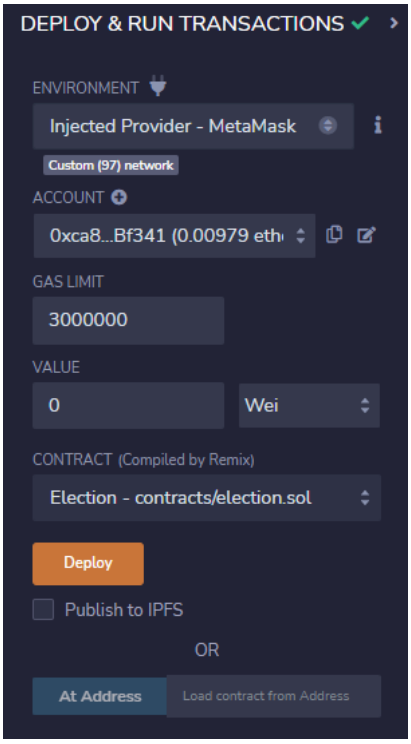

- Bytecode :

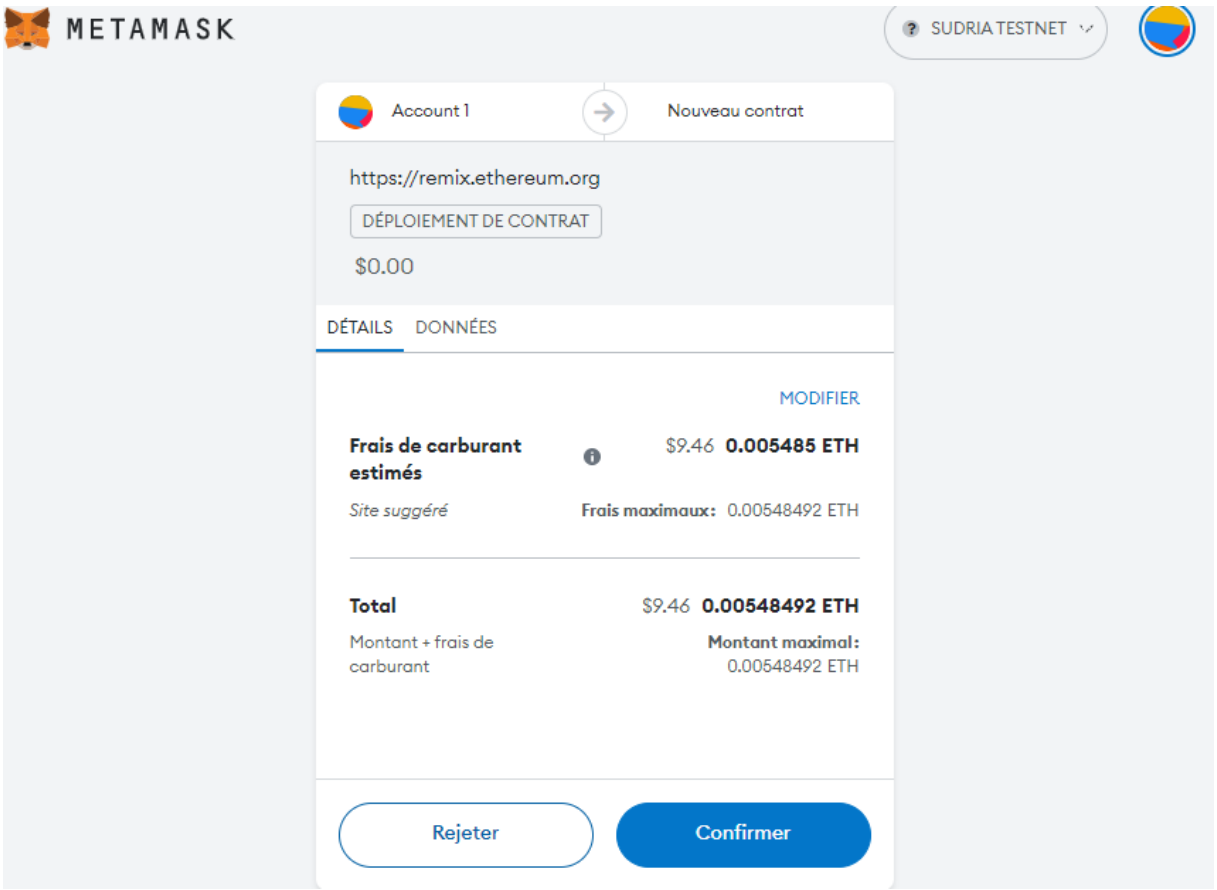

```
608060405233600806101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff
160217905550610897806100536000396000f3006080604052600436106100835760003557c01000000000000000000
0000000000000000000000000000000000000000900463ffffffff1680630121b93f146100885780632d35a8a2146100b55
780633477ee2e146100e0578063462e91ec146101945780638da5cb5b146101fd578063a3ec138d14610254578063f2
fde38b146102af575b600080fd5b34801561009457600080fd5b506100b360048036038101908080359060200190929
1905050506102f2565b005b3480156100c157600080fd5b506100ca610415565b604051808281526020019150506040
5180910390f35b3480156100ec57600080fd5b5061010b60048036038101908080359060200190929190505050610410
```

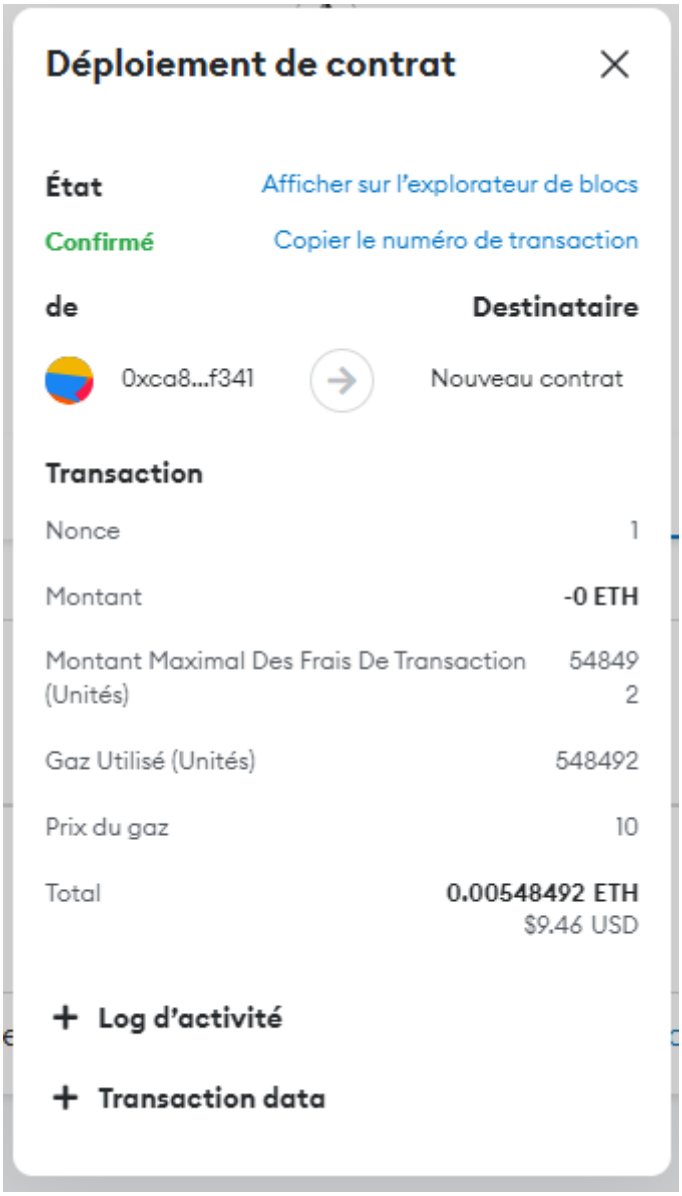
b565b60405180848152602001806020018381526020018281038252848181518152602001915080519060200190808
38360005b838110156101575780820151818401526020810190506101013c565b50505050905090810190601f1680156
1018457808203805160018360200361010000a031916815260200191505b509450505050506040518091039 0f35b348
0156101a057600080fd5b506101fb600480360381019080803590602001908201803590602001908080601f01602080
91040260200160405190810160405280939291908181526020018383808284378201915050505050509192919290 50
50506104dd565b005b3480156102095760 0080fd5b506102126105 5a565b604051808273fffffffffffffffffffffffffffffffffffff
fff1673ffffffffffffffffffffffffffffffffffffffff16815260200191505060405180910390f35b348015610260576 00080fd5b50610
2956004803603810190808035 73ffffffffffffffffffffffffffffffffffffffff169060200190929190505050610 57f565b604051808
2151515158152602001915050604051809103 90f35b3480156102bb57600080fd5b506102f06004803603810190808 0
3573ffffffffffffffffffffffffffffffffffffffff16906020019092919050505061059f565b005b600160003373ffffffffffffffffffffffffffff
ffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152602001600020600090549061010 00a900460ff161
5151561034b57600080fd5b600081118015610 35d57506003548111155b151561036857600080fd5b6001806000337
3ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152602001600020600061 01000
a81548160ff021916908315150217905550600260008281526020019081526020016000206002016000815480929190
600101919050555080 7ffff3c900d938d21d0990d786e819f29b8d05c1ef587b462b939609625b684b16604051604051
80910390a250565b60035481565b6002602052806000526040600020600091509050806000015490806001018054 60
018160011615610100020316600290048060 1f01602080910402602001604051908101604052809291908181526020
0182805460018160011615610100020316600290048015 6104cd5780601f106104a2576101008083540402835291 60
2001916104cd565b820191906000526020600020905b8154815290600101906020018083116104b057829003601f16
8201915b505050505050908060020154905083565b6003600081548092919060010191905055506060604051908101 60
40528060035481526020018281526020016000815250600260006003548152602001908152602001600020600082 01
5181600001556020820151816001019080519060200190610549929190610 7c6565b5060408201518160020155905 0
5050565b600080905490610 1000a900473ffffffffffffffffffffffffffffffffffffffff1681565b60016020528060005260406000 20
600091505490610 1000a900460ff1681565b600080905490610 1000a900473ffffffffffffffffffffffffffffffffffffffff1673ffffffff
ffffffffffffffffffffffffffffffff163373ffffffffffffffffffffffffffffffffffffffff1614151561066357604051 7f08c379a00000000000000
000000000000000000000000000000000000000008152600401808060200182810382526018815260200180 7f4e6f
7420617574686f72697a6564206f7065726174696f6e000000000000000081525060200191505060405180910390fd5
b600073ffffffffffffffffffffffffffffffffffffffff168173ffffffffffffffffffffffffffffffffffffffff161415151561070857604051 7f08c379a
000000000000000000000000000000000000000000000000000000008152600401808060200182810382526019815
2602001807f4164647265737320736f756c646e2774206265207a65726f00000000000000815250602001915050 60
405180910390fd5b8073ffffffffffffffffffffffffffffffffffffffff16600080905490610 1000a900473ffffffffffffffffffffffffffffffffffff
fff1673ffffffffffffffffffffffffffffffffffffffff167f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e
0604051604051809103 90a3806000806101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffff
ffffffffffffffffffff1602179055505050565b82805460018160011615610100020316600290049060005260206000209060
1f016020900481019282601f1061080757805160ff191683800117855556108355 65b828001600101855558215610835 5
79182015b828111156108345782518255916020019190600101906108195 65b5b509050610842919061084656 5b509
0565b61086891905b808211156108645760008160009055506001016108 4c565b5090565b905600a165627a7a72305
82091c0a3a641f272a13d7a4b87425b46d8f1a9570cb860be4b6b16d63f55b913950029

m)

- <u>Déploiement du smart contract Election.sol :</u>


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Provider - MetaMask
Custom (97) network
ACCOUNT
0xca8...Bf341 (0.00979 eth
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT (Compiled by Remix)
Election - contracts/election.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address


METAMASK
SUDRIA TESTNET
Account 1
Nouveau contrat
https://remix.ethereum.org
DÉPLOIEMENT DE CONTRAT
$0.00
DÉTAILS
DONNÉES
MODIFIER
Frais de carburant estimés
$9.46 0.005485 ETH
Site suggéré
Frais maximaux: 0.00548492 ETH
Total
$9.46 0.00548492 ETH
Montant + frais de carburant
Montant maximal: 0.00548492 ETH
Rejeter
Confirmer



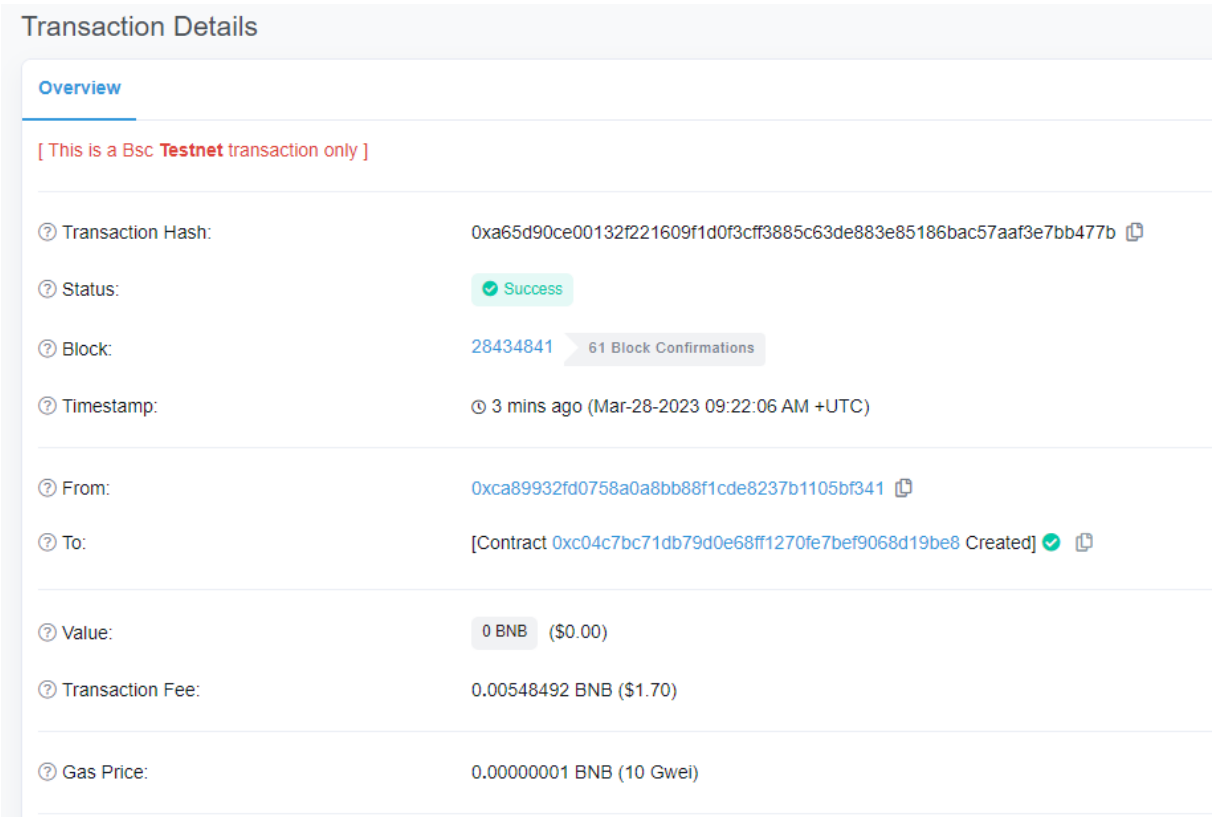
- <u>Justifications des frais de transaction :</u>

Lorsque l'on déploie un contrat sur la blockchain, les frais de transaction sont généralement plus élevés que lors de l'envoi d'ETH à une adresse car la création d'un nouveau contrat nécessite plus de ressources et de traitement par les mineurs du réseau. Cette opération est plus complexe et nécessite davantage de puissance de calcul pour vérifier les termes et les conditions du contrat.

D'autre part, lors de l'envoi d'ETH à une adresse, les frais de transaction sont généralement moins élevés car il s'agit d'une opération plus simple qui nécessite moins de ressources et de temps de traitement pour être vérifiée et validée par les mineurs du réseau.

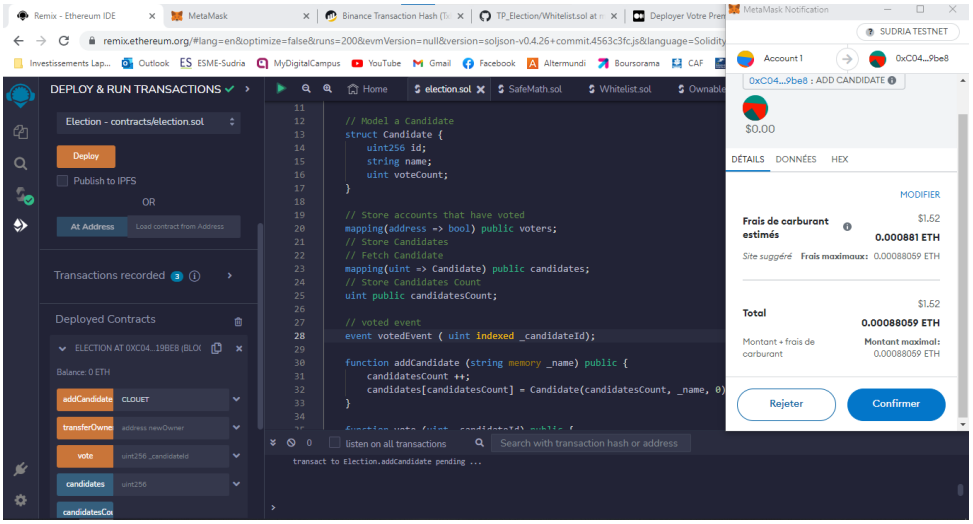
- Adresse publique du smart contract :

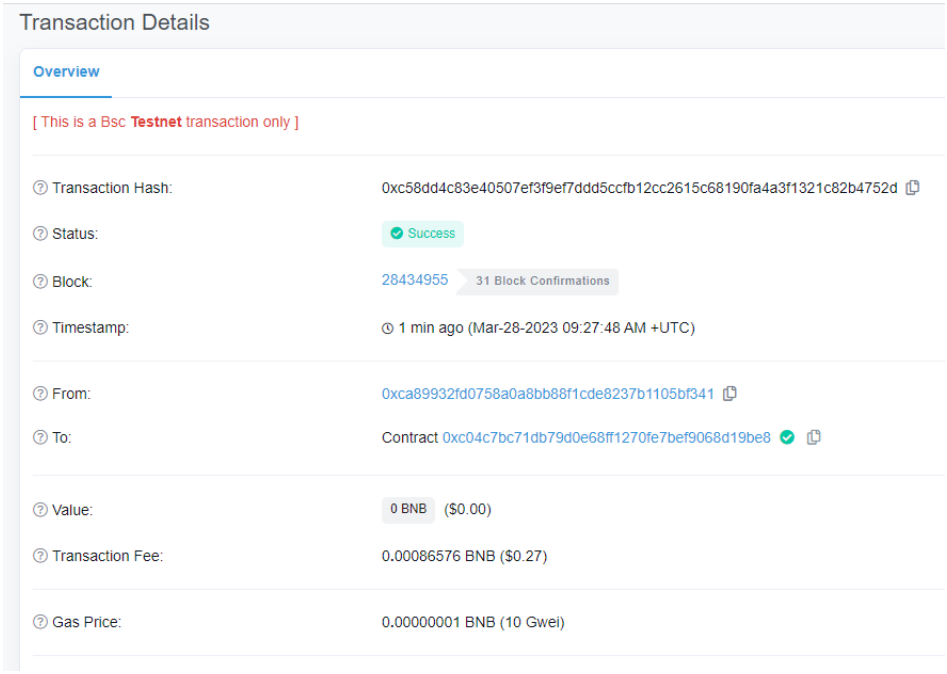
0xa65d90ce00132f221609f1d0f3cff3885c63de883e85186bac57aaf3e7bb477b



n)

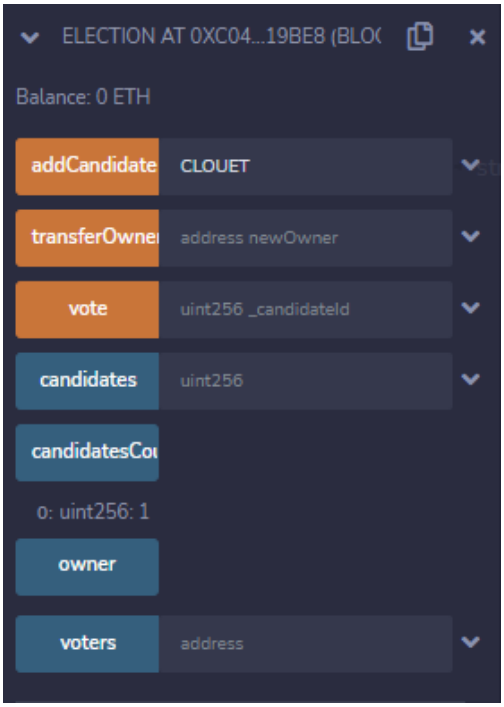
- <u>Ajout du premier candidat sur le smart contract :</u>





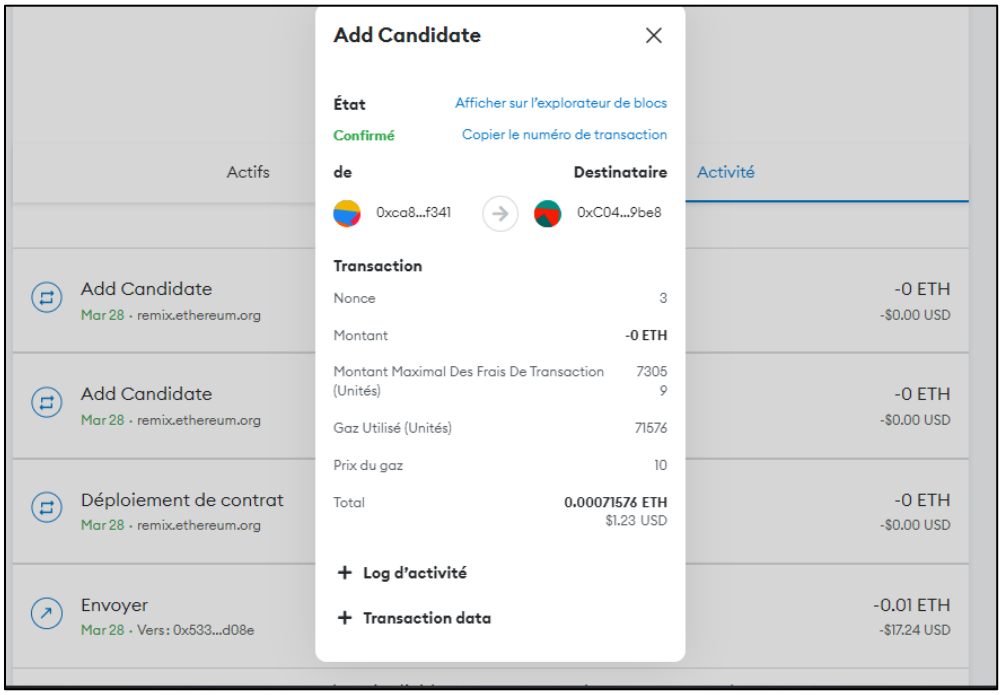
p)

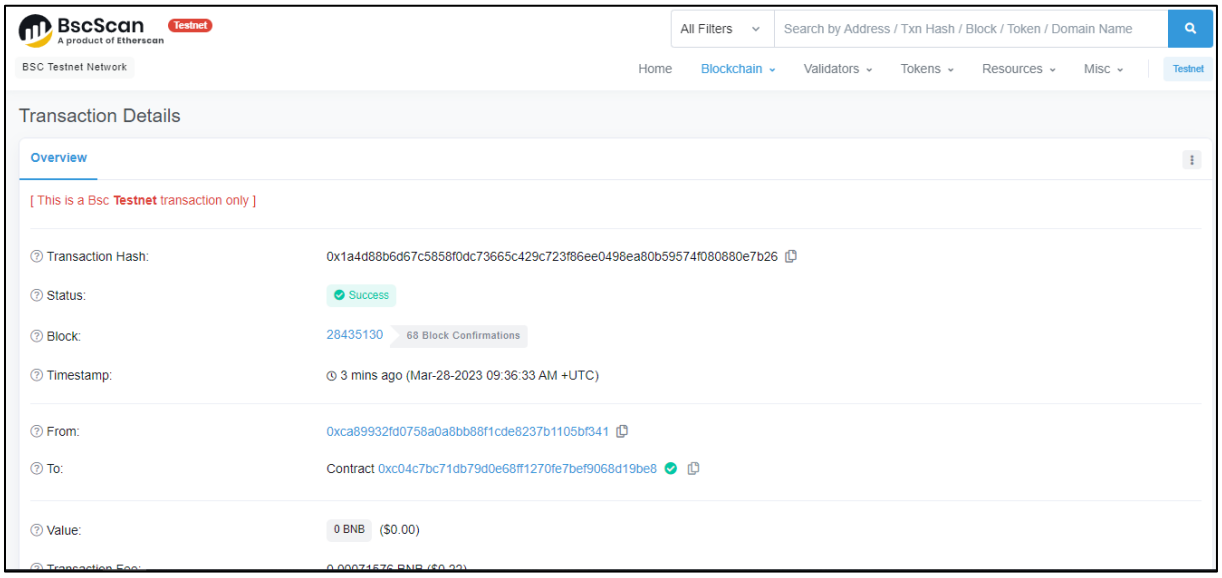
Candidate id : 1



q)

- <u>Ajout du second candidat :</u>

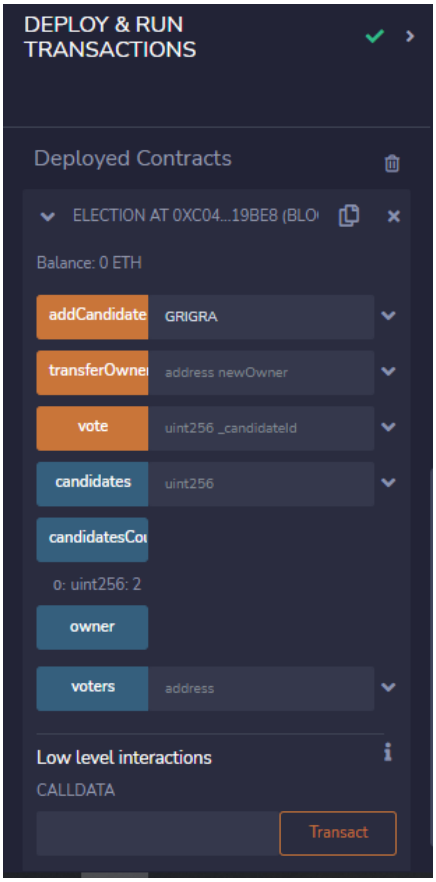




r)

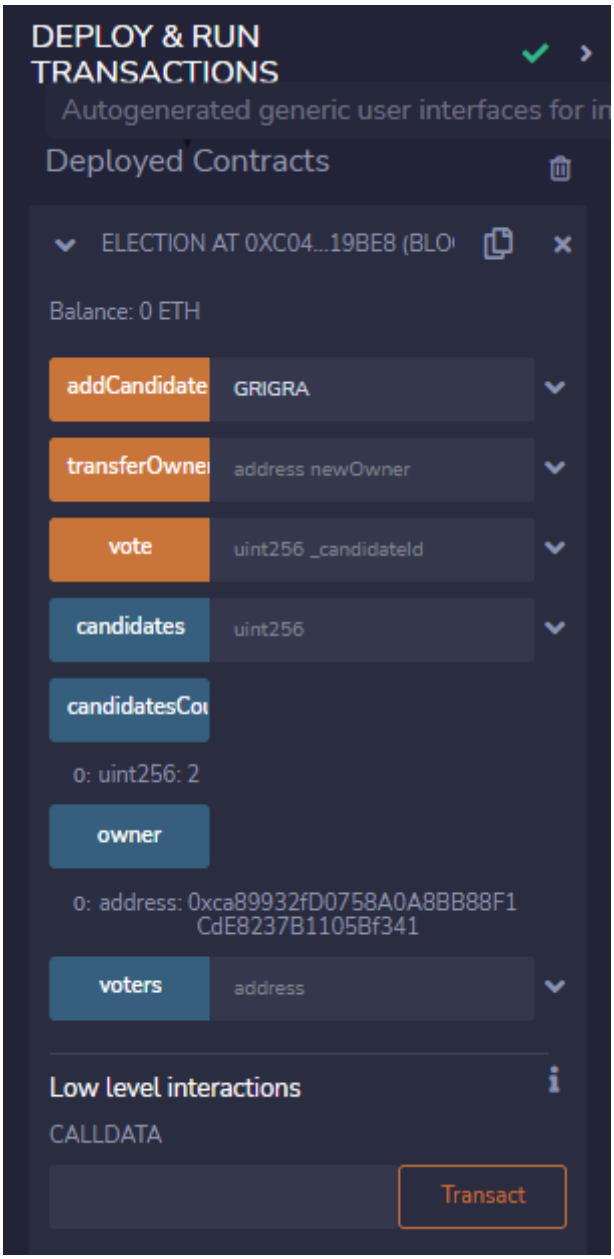
- Valeur du second candidat :

  Candidate id : 2



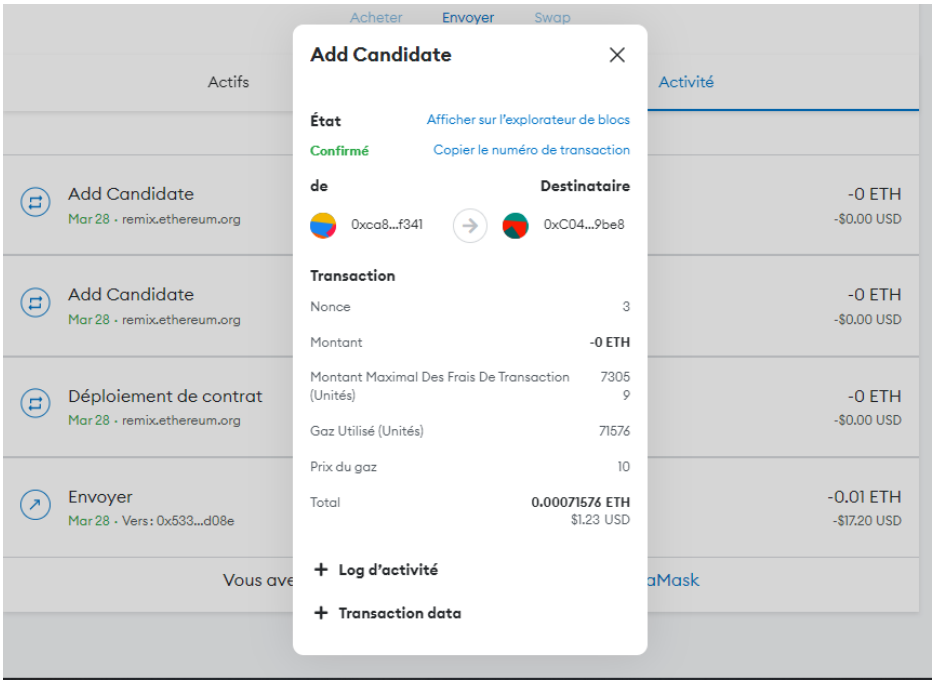
s)

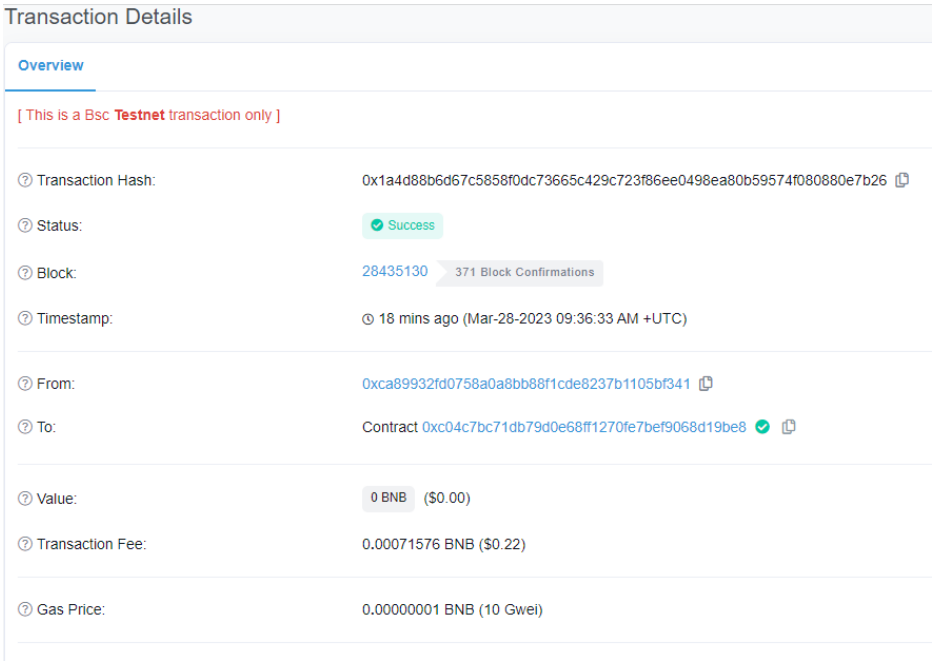
- Adresse du propriétaire du contrat : Moi-même (voir pj ci-dessous)



t)

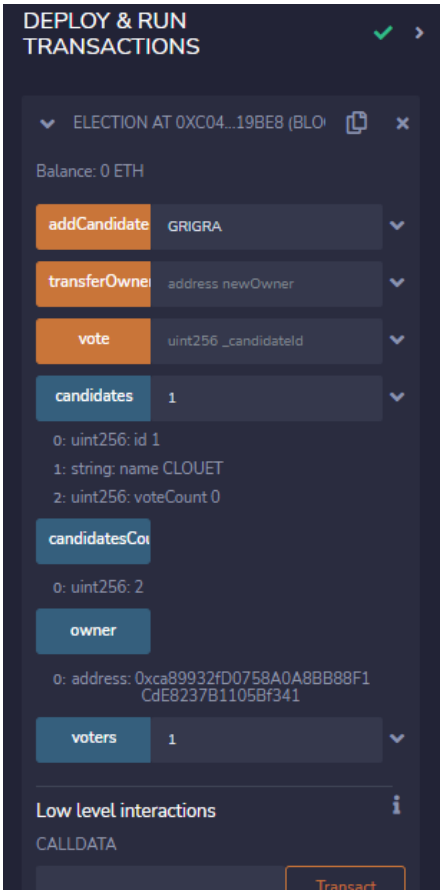
- <u>Vote pour l'un des candidats :</u>





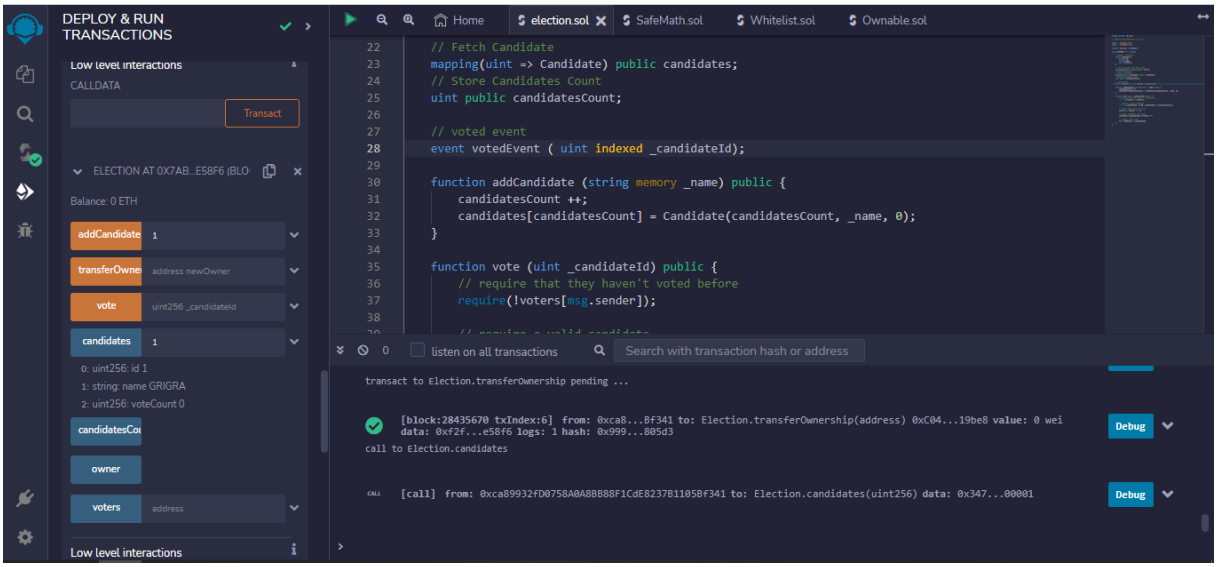
u)

- <u>Prise en compte du vote :</u>

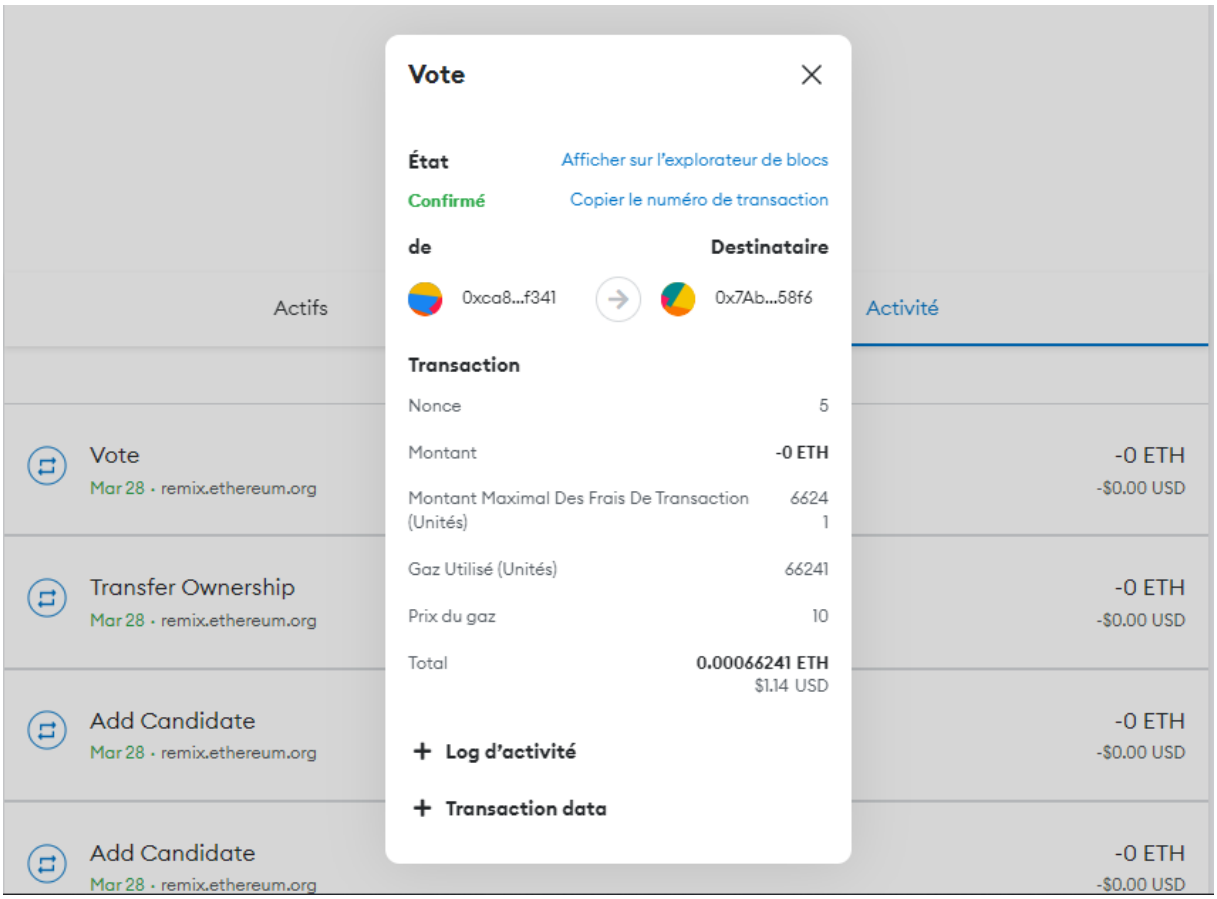


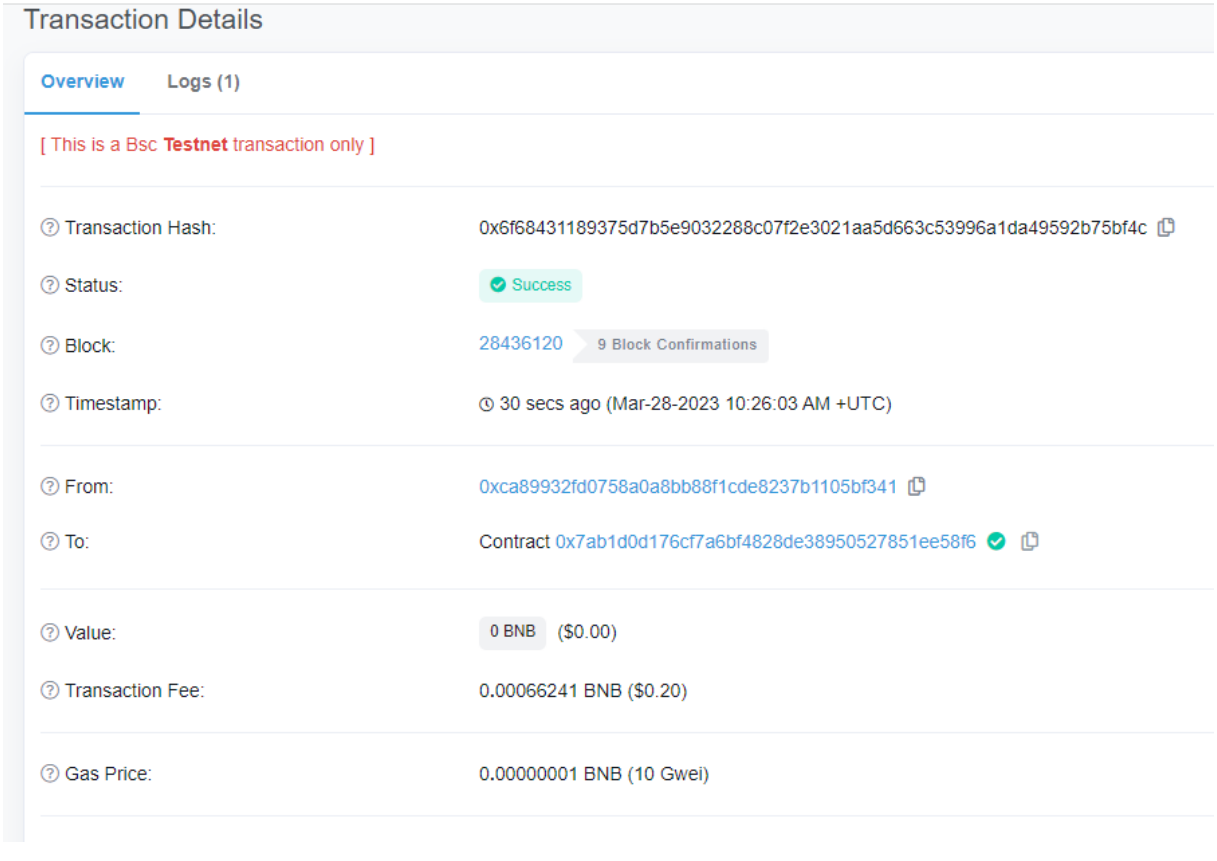
v)

- Clé publique du contrat d'Adam : 0x7Ab1d0D176CF7A6BF4828dE38950527851eE58f6

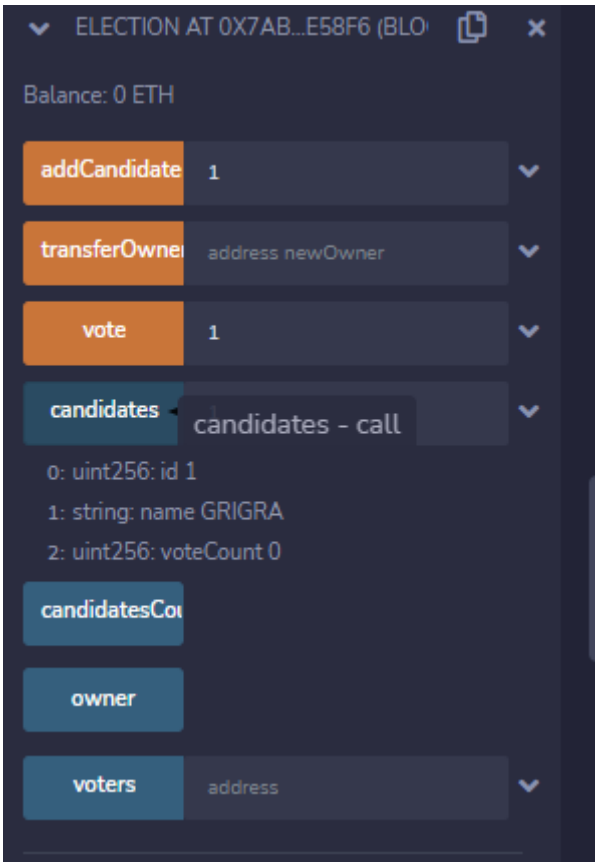


Vote que j'ai fait chez Adam :

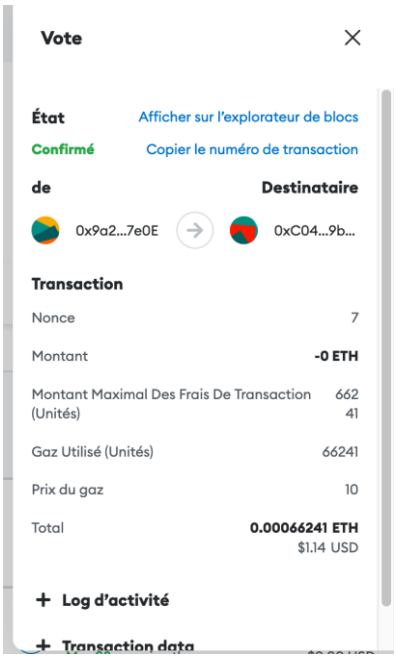




Vote pour le candidat 1 (GRIGRA) :

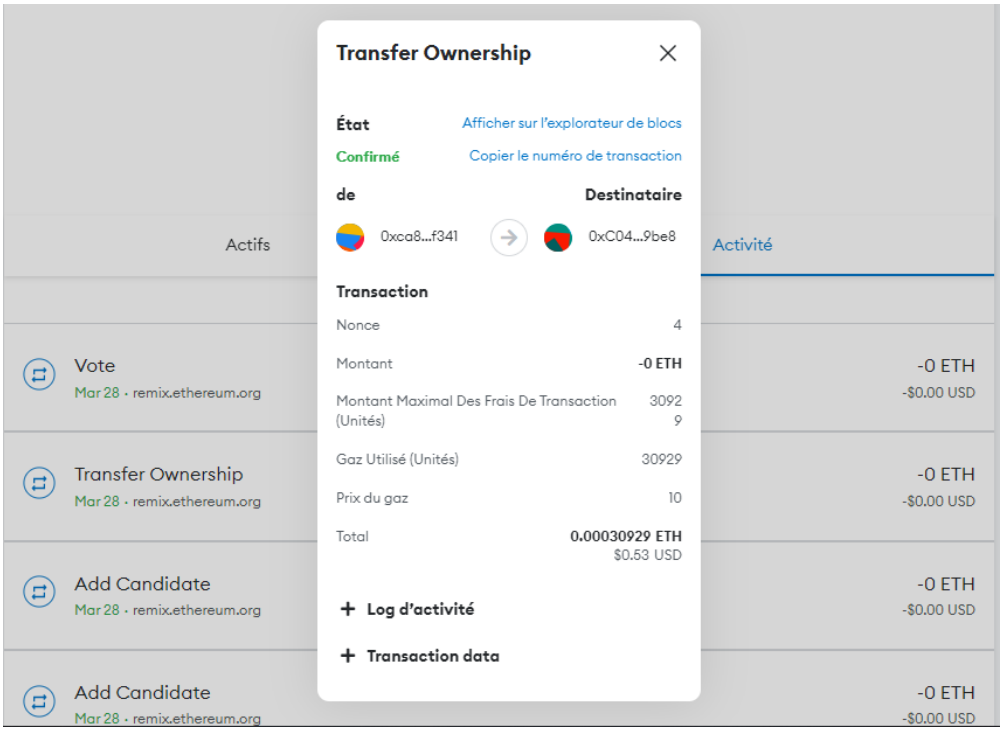


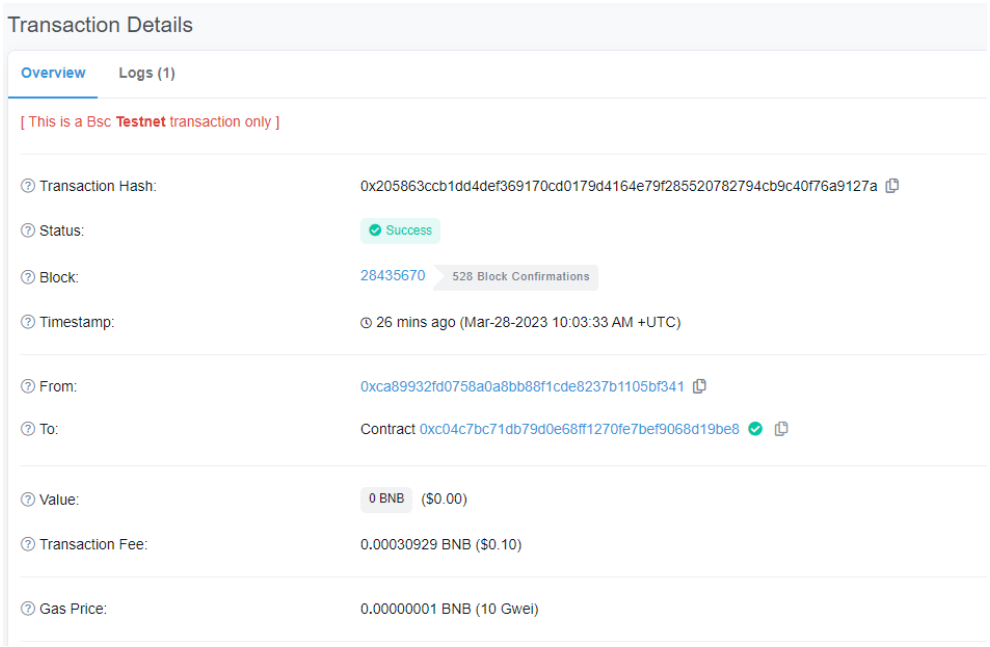
Vote d'Adam sur mon contrat :



w)

- <u>Transfert de la propriété à mon camarade :</u>





x)

Il faudrait créer une variable prioritaire sur le même principe qu'un rôle administrateur qui aurait pour rôle de stocker l'identifiant prioritaire. Ensuite, il faut initialiser cette variable avec la valeur choisie.

Nous pourrions ensuite modifier la fonction "addCandidate" en y ajoutant une vérification d'authentification à l'aide d'un id et d'un password. A l'aide d'une instruction "require" nous pourrions vérifier que les identifiants sont corrects.

Il ne manque plus qu'à déployer une seconde fois le contrat pour qu'il soit actif.

y)

- Voici ce qu'il faut changer dans le code pour faire en sorte que je sois le seul à pouvoir ajouter un nouveau candidat :

```
function addCandidate (string memory _name) public onlyOwner {
    candidatesCount ++;
    candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
}
```