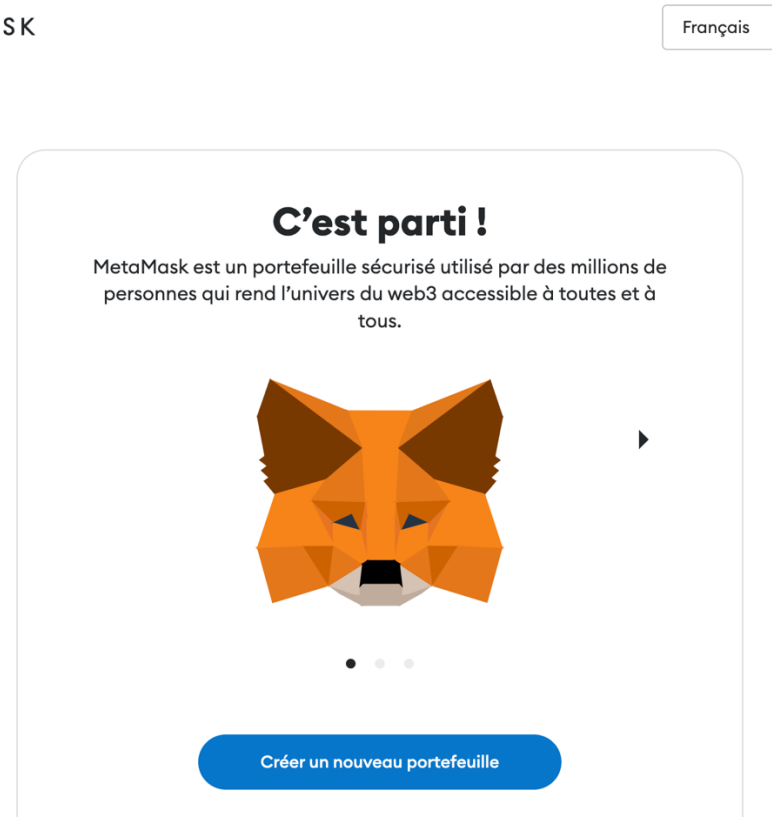
TP1 : Blockchain – Matthieu CHILA – A2MSI

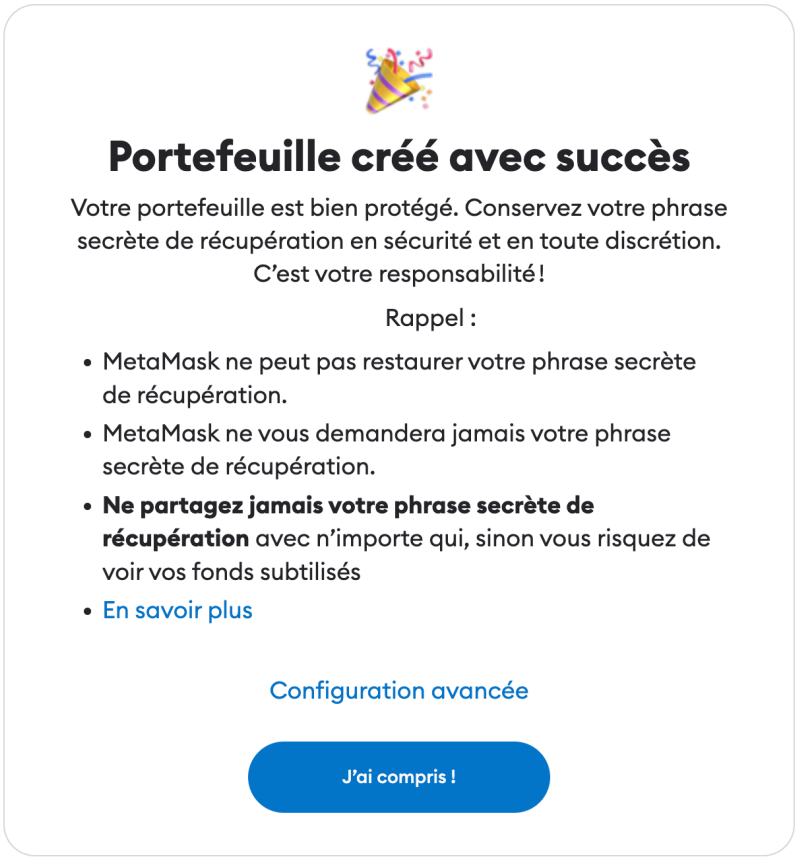
Développer, Deloyer et Interagir avec un contrat intelligent sur Ethereum

1. Prise en main des outils Remix et Metamask

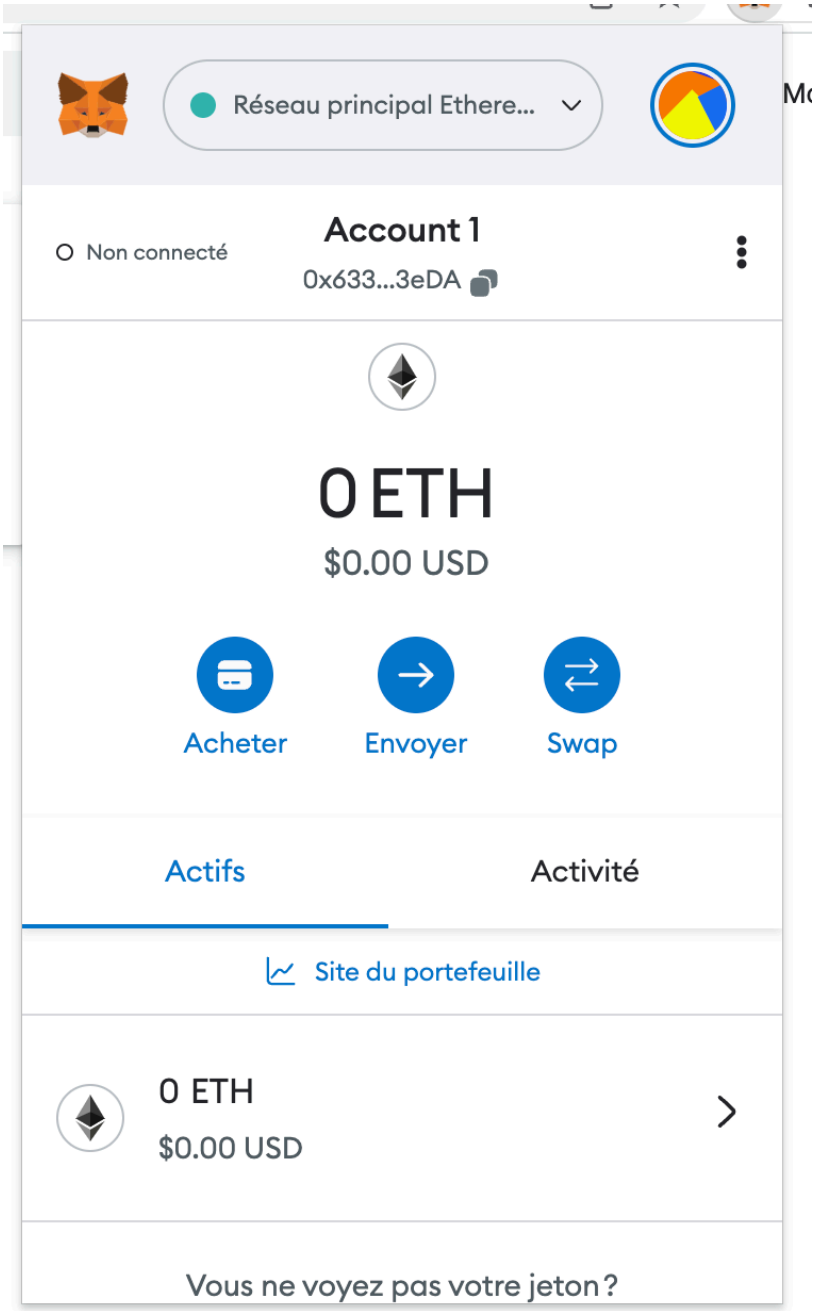
A)



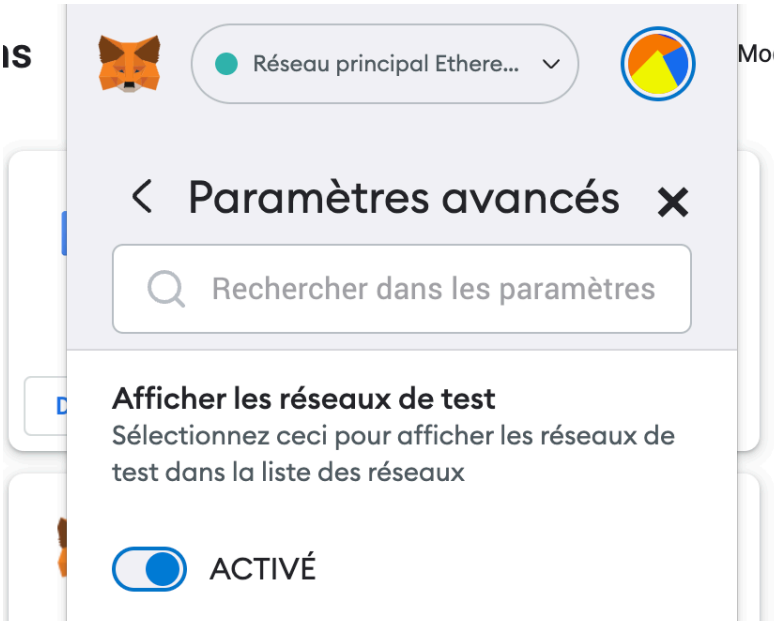
B)

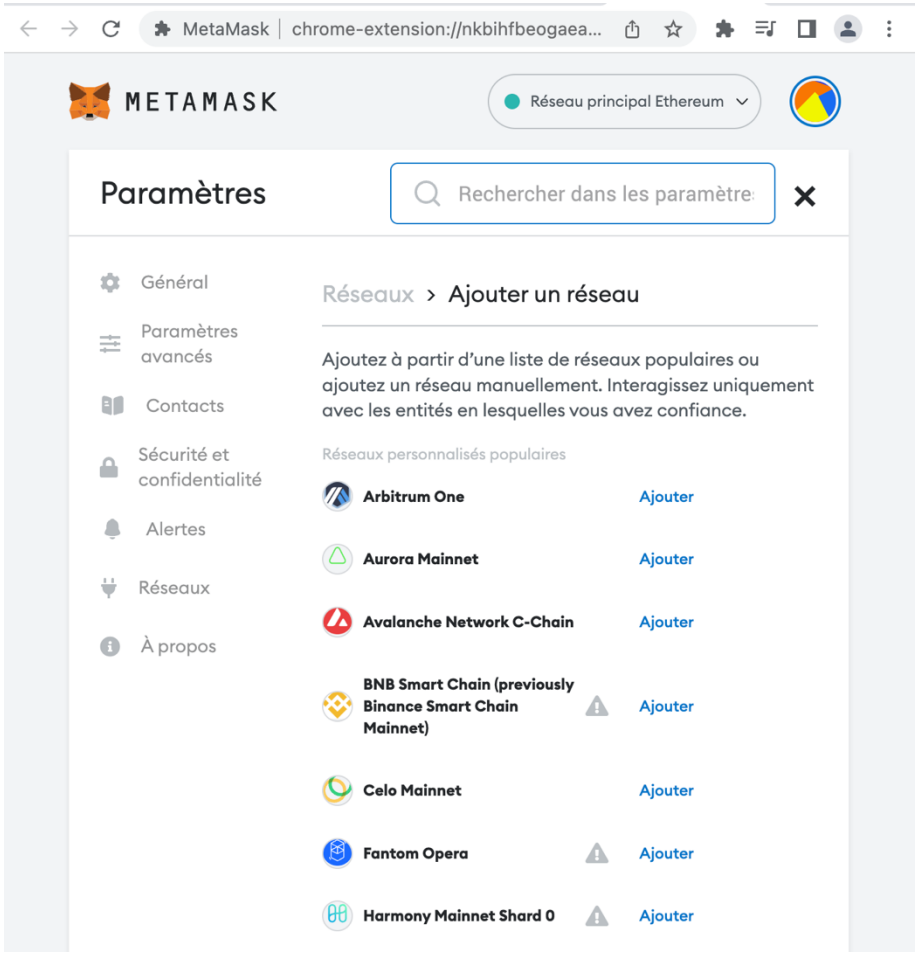


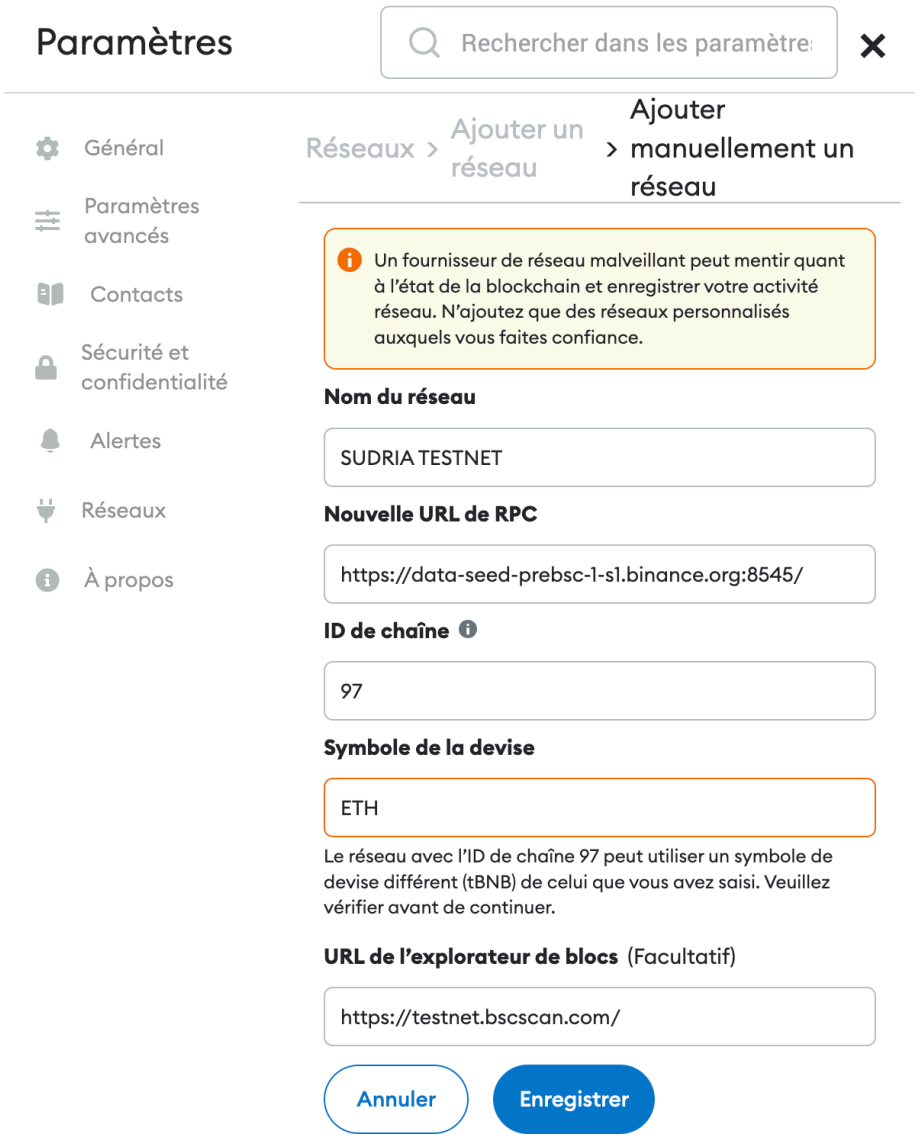
C) Clé publique : 0x63356ab45f581B4eDE4519a118ecD55F1E463eDA



D)

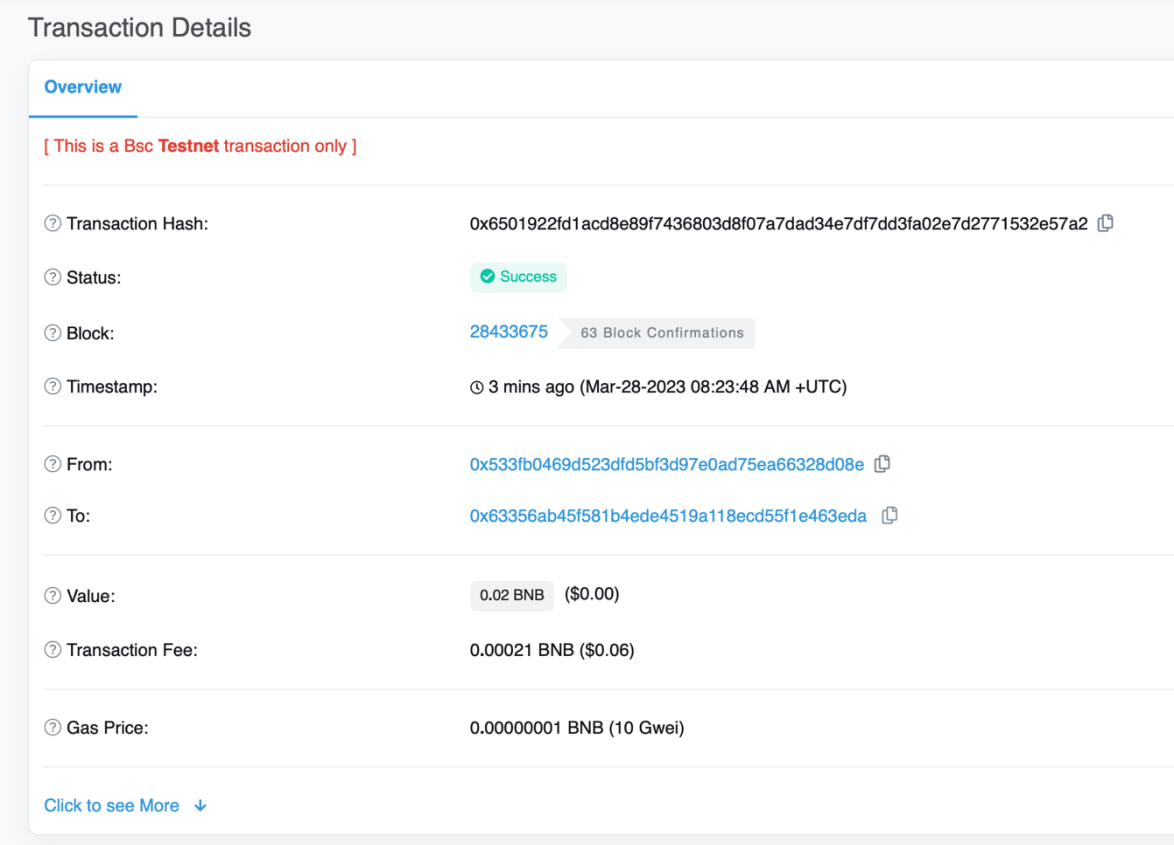



MetaMask | chrome-extension://nkbihfbeogaea...
METAMASK
Réseau principal Ethereum
Paramètres
Rechercher dans les paramètre
Général
Paramètres avancés
Contacts
Sécurité et confidentialité
Alertes
Réseaux
À propos
Réseaux > Ajouter un réseau
Ajoutez à partir d'une liste de réseaux populaires ou ajoutez un réseau manuellement. Interagissez uniquement avec les entités en lesquelles vous avez confiance.
Réseaux personnalisés populaires
Arbitrum One Ajouter
Aurora Mainnet Ajouter
Avalanche Network C-Chain Ajouter
BNB Smart Chain (previously Binance Smart Chain Mainnet) Ajouter
Celo Mainnet Ajouter
Fantom Opera Ajouter
Harmony Mainnet Shard 0 Ajouter

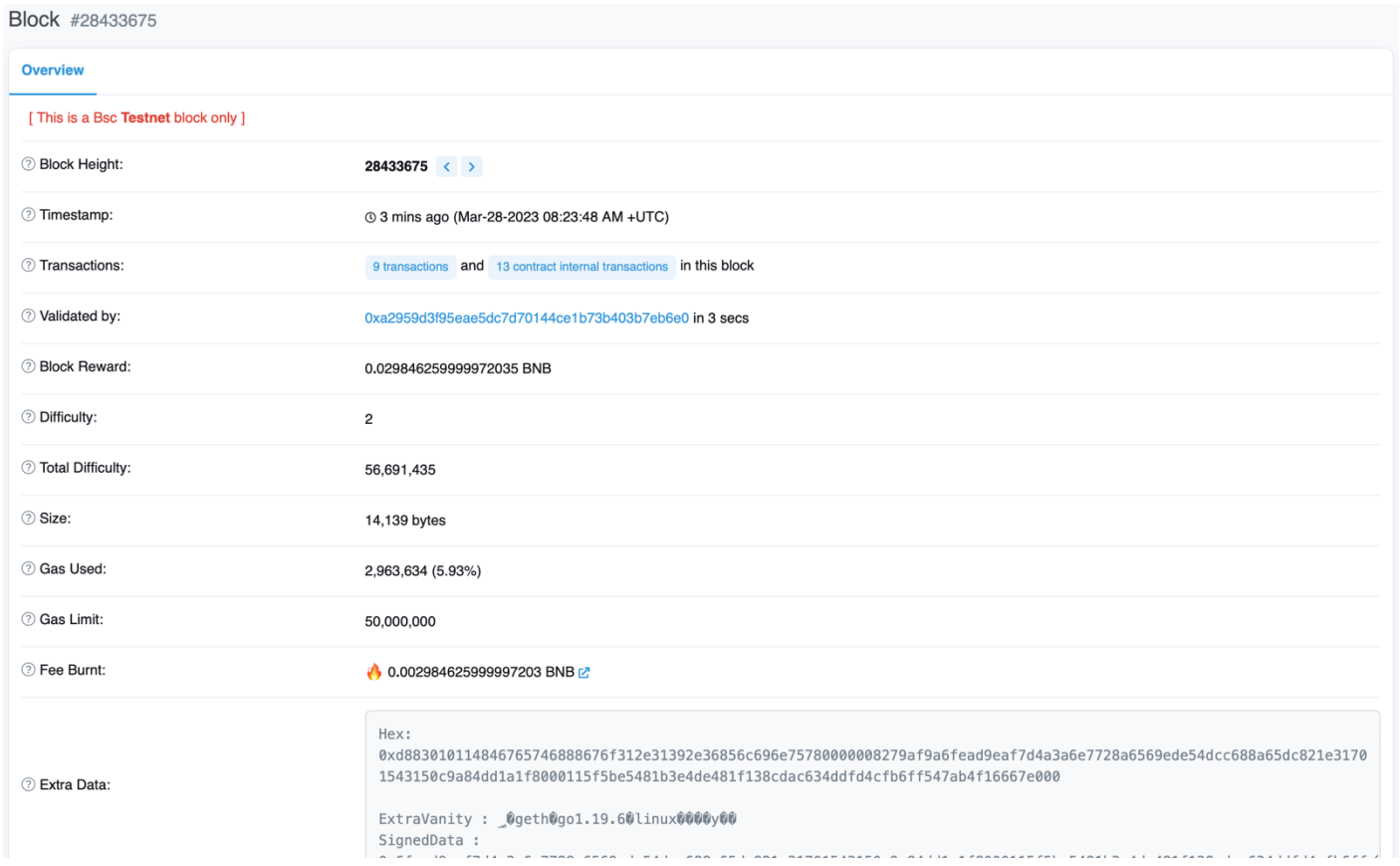

Paramètres
Rechercher dans les paramètre
Général
Paramètres avancés
Contacts
Sécurité et confidentialité
Alertes
Réseaux
À propos
Réseaux > Ajouter un réseau > Ajouter manuellement un réseau
Un fournisseur de réseau malveillant peut mentir quant à l'état de la blockchain et enregistrer votre activité réseau. N'ajoutez que des réseaux personnalisés auxquels vous faites confiance.
Nom du réseau
SUDRIA TESTNET
Nouvelle URL de RPC
https://data-seed-prebsc-1-s1.binance.org:8545/
ID de chaîne
97
Symbole de la devise
ETH
Le réseau avec l'ID de chaîne 97 peut utiliser un symbole de devise différent (tBNB) de celui que vous avez saisi. Veuillez vérifier avant de continuer.
URL de l'explorateur de blocs (Facultatif)
https://testnet.bscscan.com/
Annuler
Enregistrer

E) Obtention de nos premiers ETH.

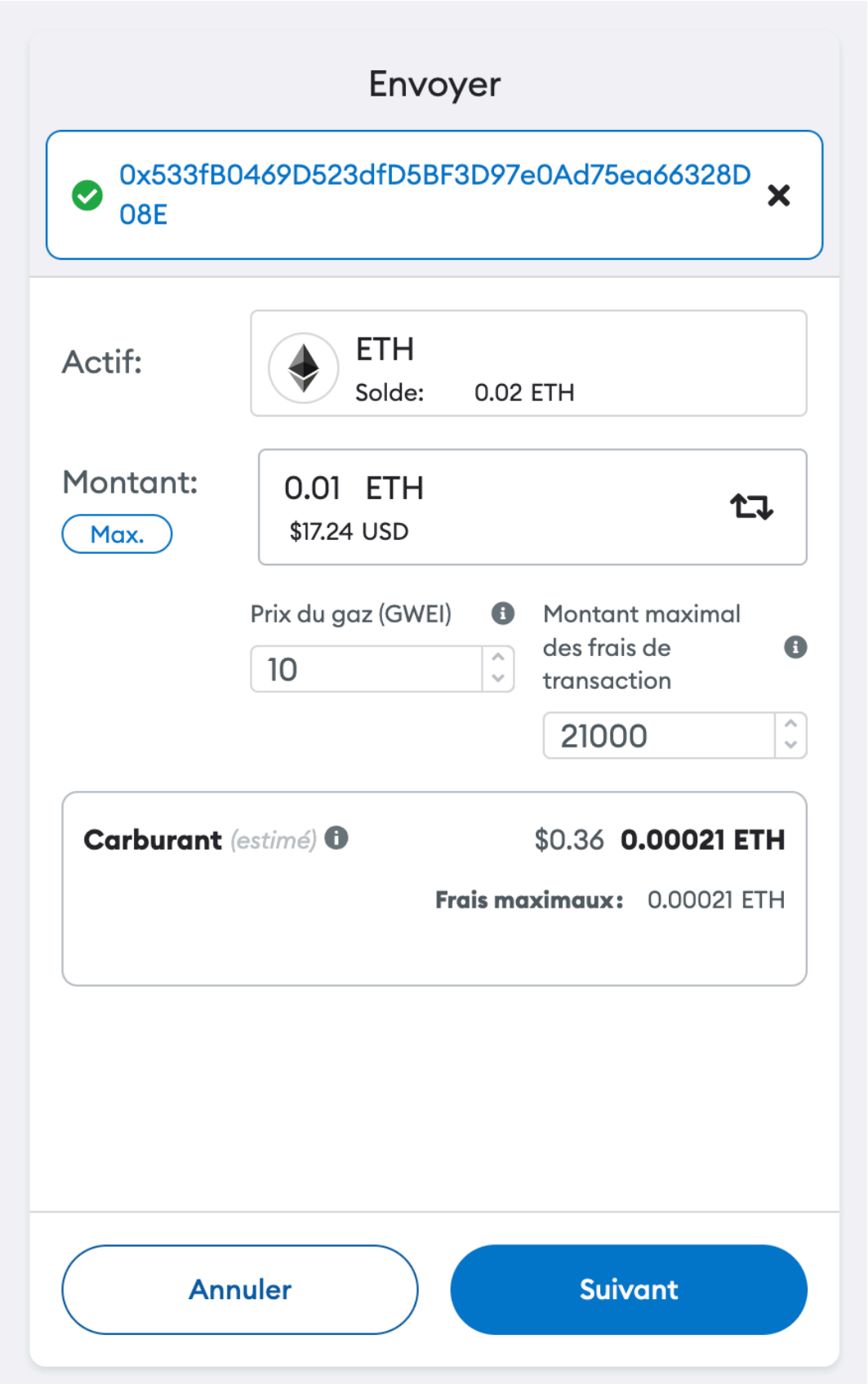
F) Transaction générée vers mon compte :

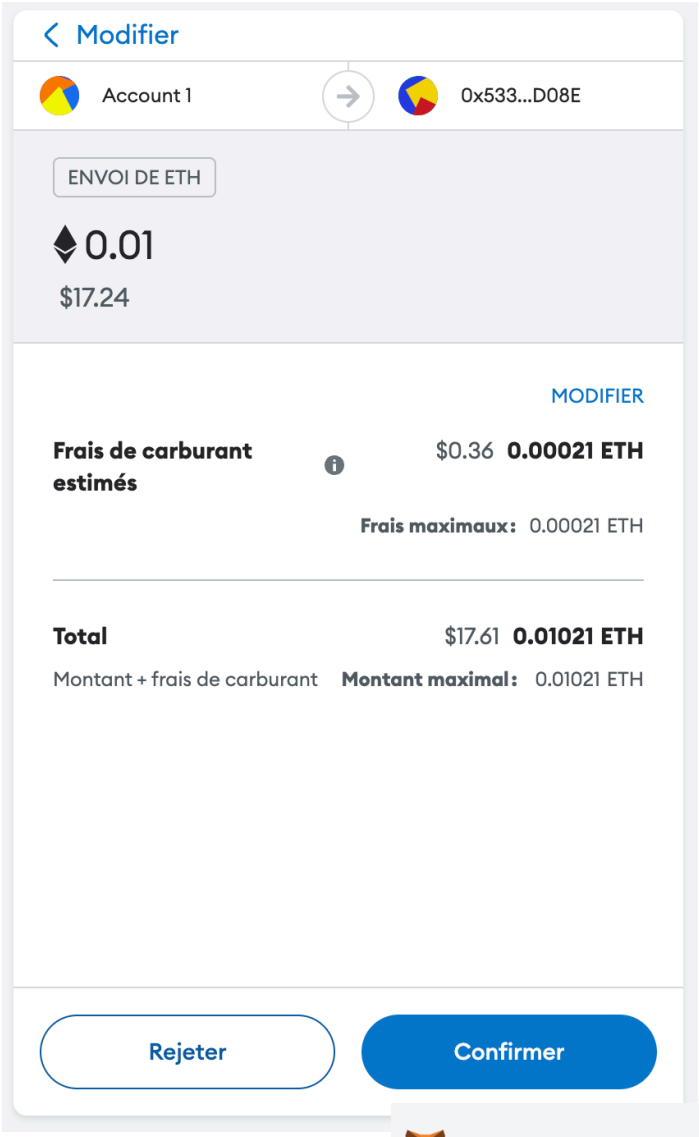


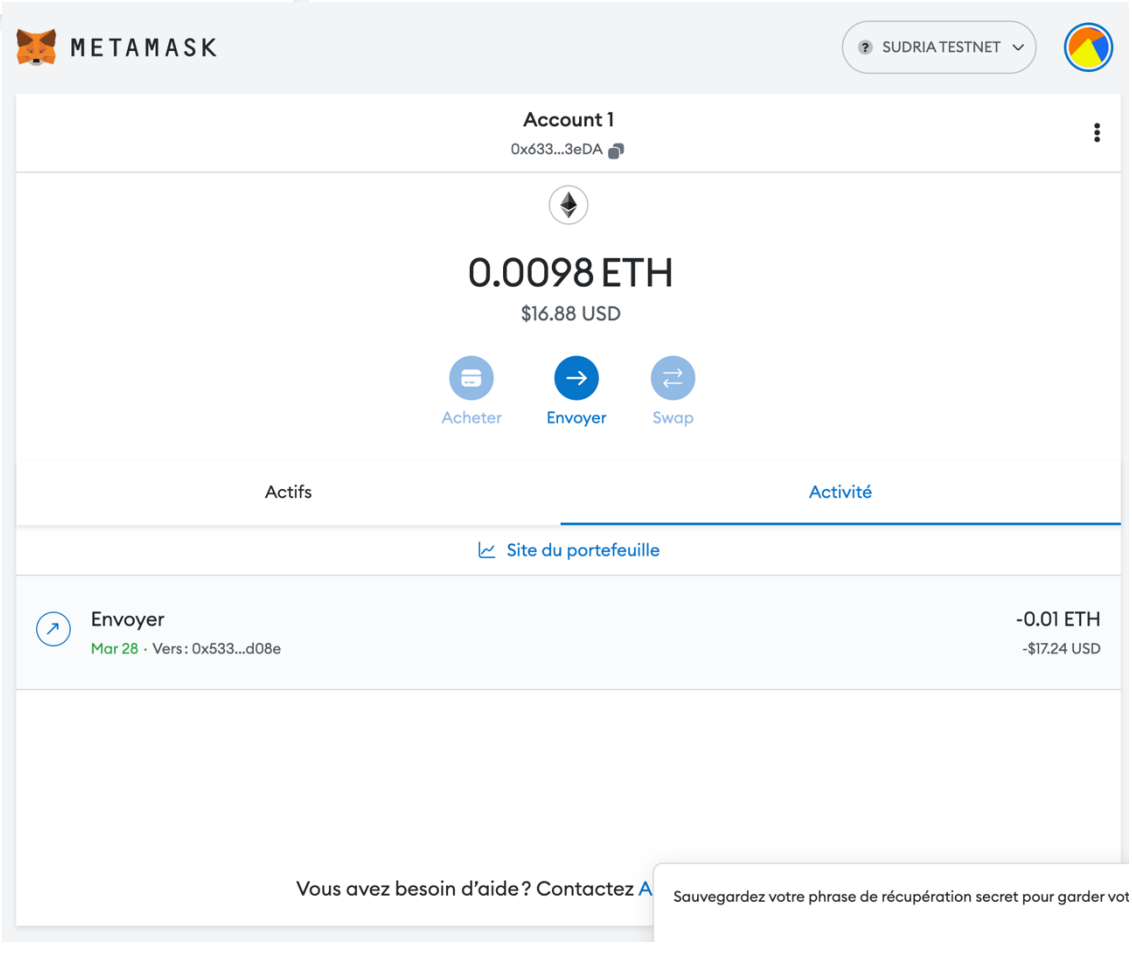
G) Concernant le block :



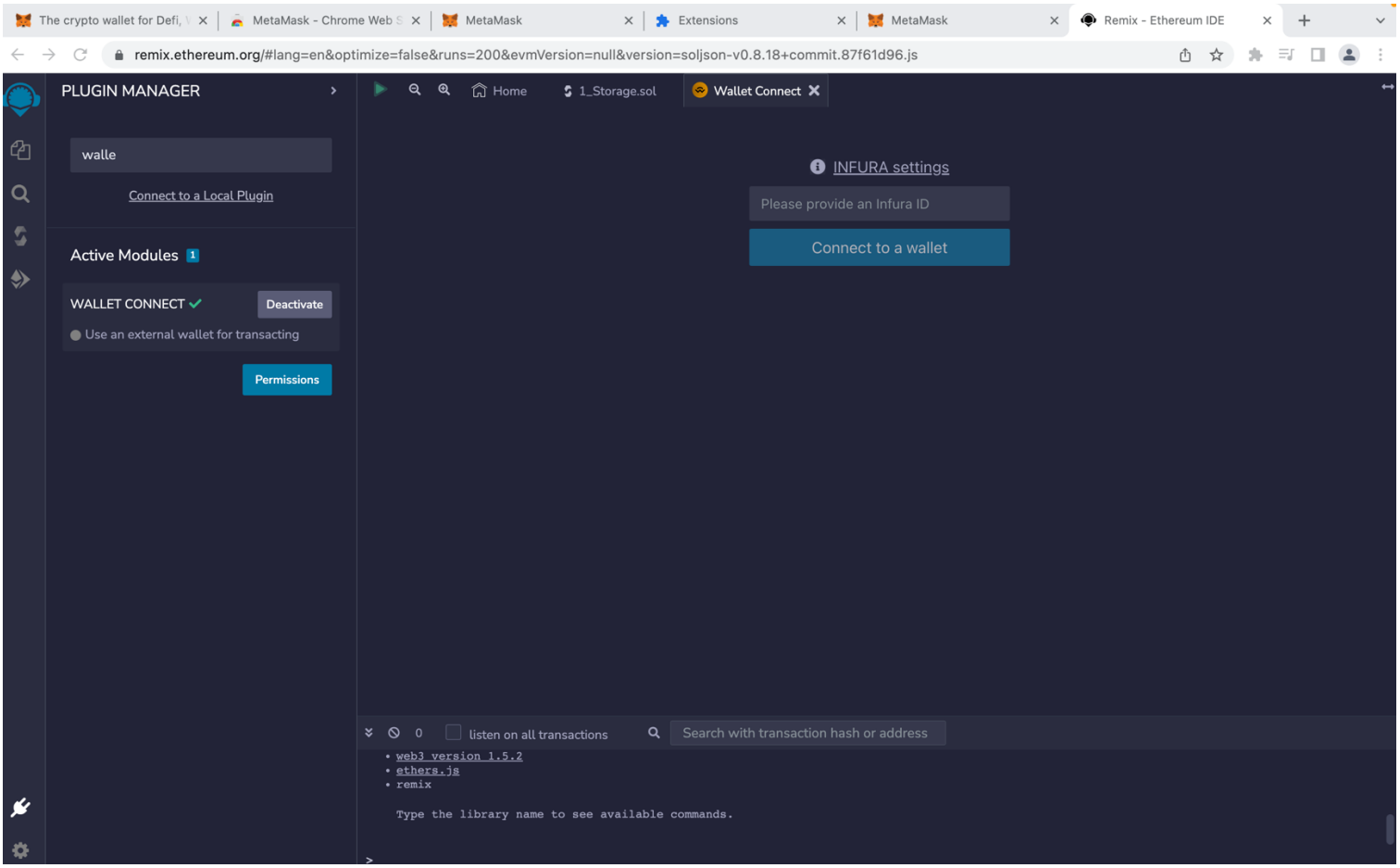
H) Première transaction Ethereum sur le réseau envoyant 0.01 ETH à l'adresse suivante « 0x533fB0469D523dfD5BF3D97e0Ad75ea66328D08E » :

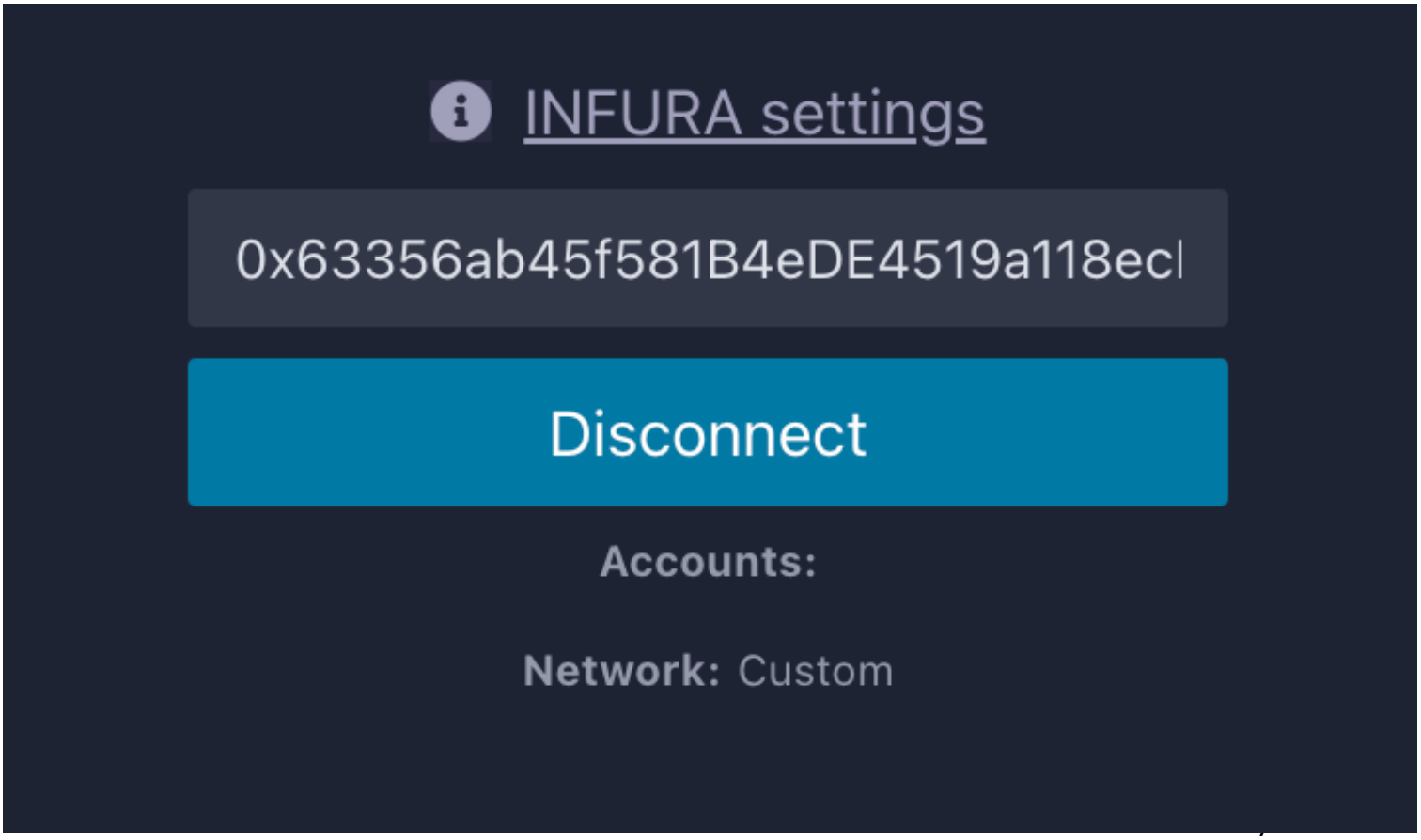



Modifier
Account 1
0x533...D08E
ENVOI DE ETH
0.01
$17.24
MODIFIER
Frais de carburant estimés
$0.36 0.00021 ETH
Frais maximaux: 0.00021 ETH
Total
$17.61 0.01021 ETH
Montant + frais de carburant
Montant maximal: 0.01021 ETH
Rejeter
Confirmer


METAMASK
SUDRIA TESTNET
Account 1
0x633...3eDA
0.0098 ETH
$16.88 USD
Acheter
Envoyer
Swap
Actifs
Activité
Site du portefeuille
Envoyer
Mar 28 · Vers: 0x533...d08e
-0.01 ETH
-$17.24 USD
Vous avez besoin d'aide? Contactez A
Sauvegardez votre phrase de récupération secret pour garder vot

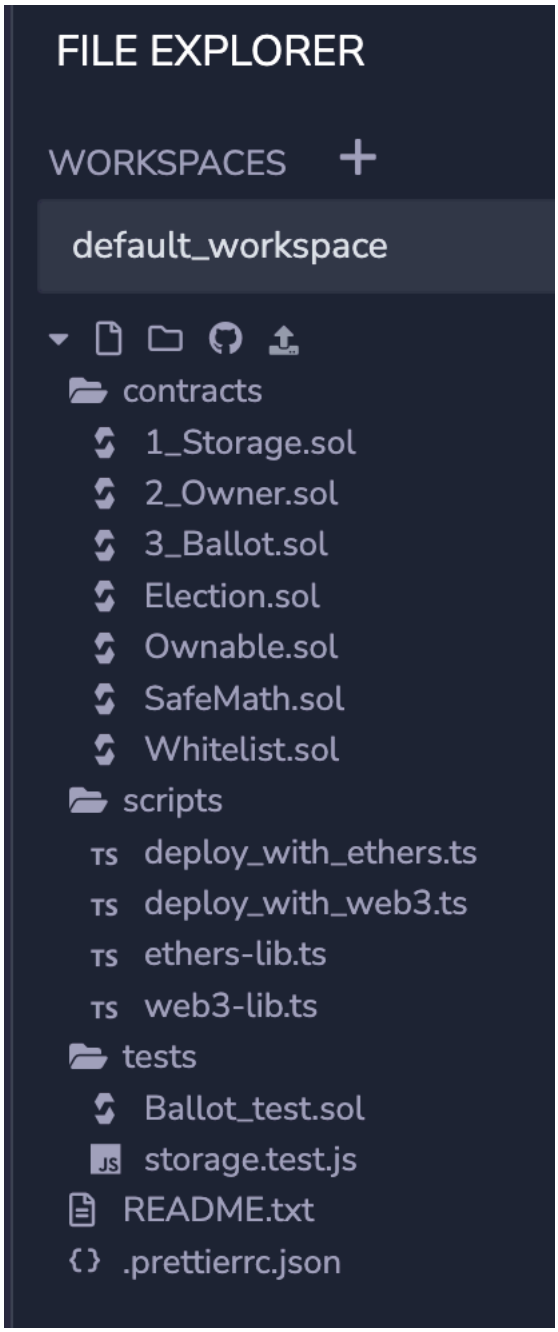
I) Ouverture de Remix :



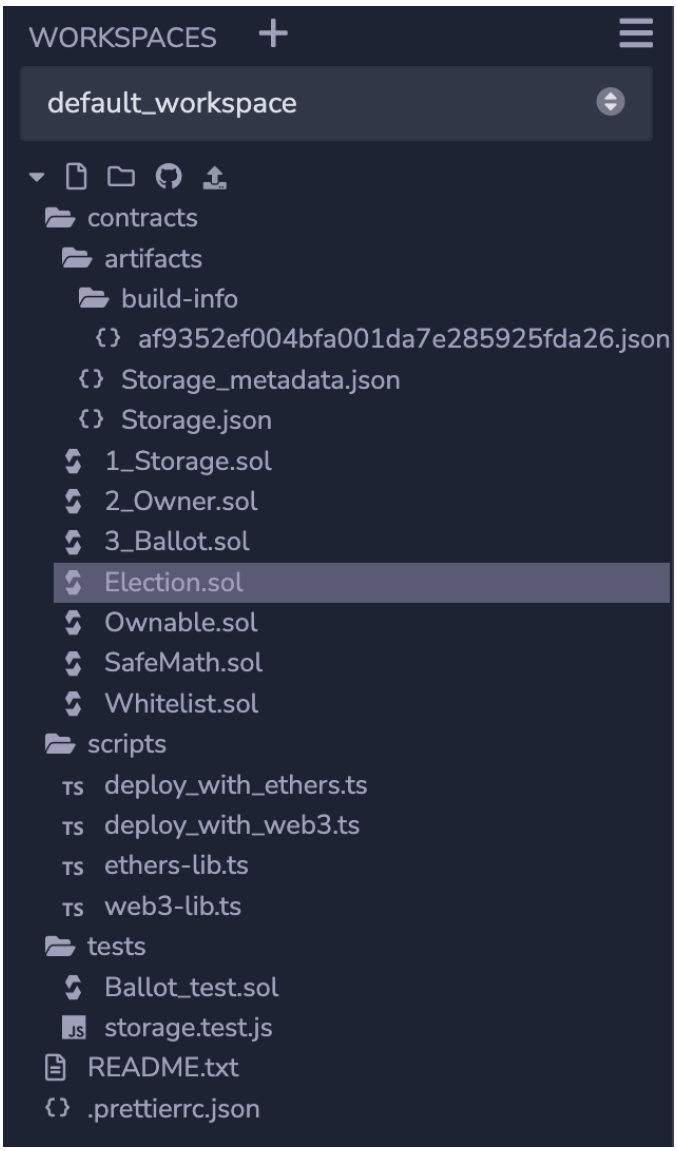


J) Récupération du code source de mon premier smart contract.

K) Ajout de l'ensemble des fichiers Solidity sur Remix :



L) Compilation sur smart contract « Election » :



ABI :


```
[
	{
		"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"
},
```

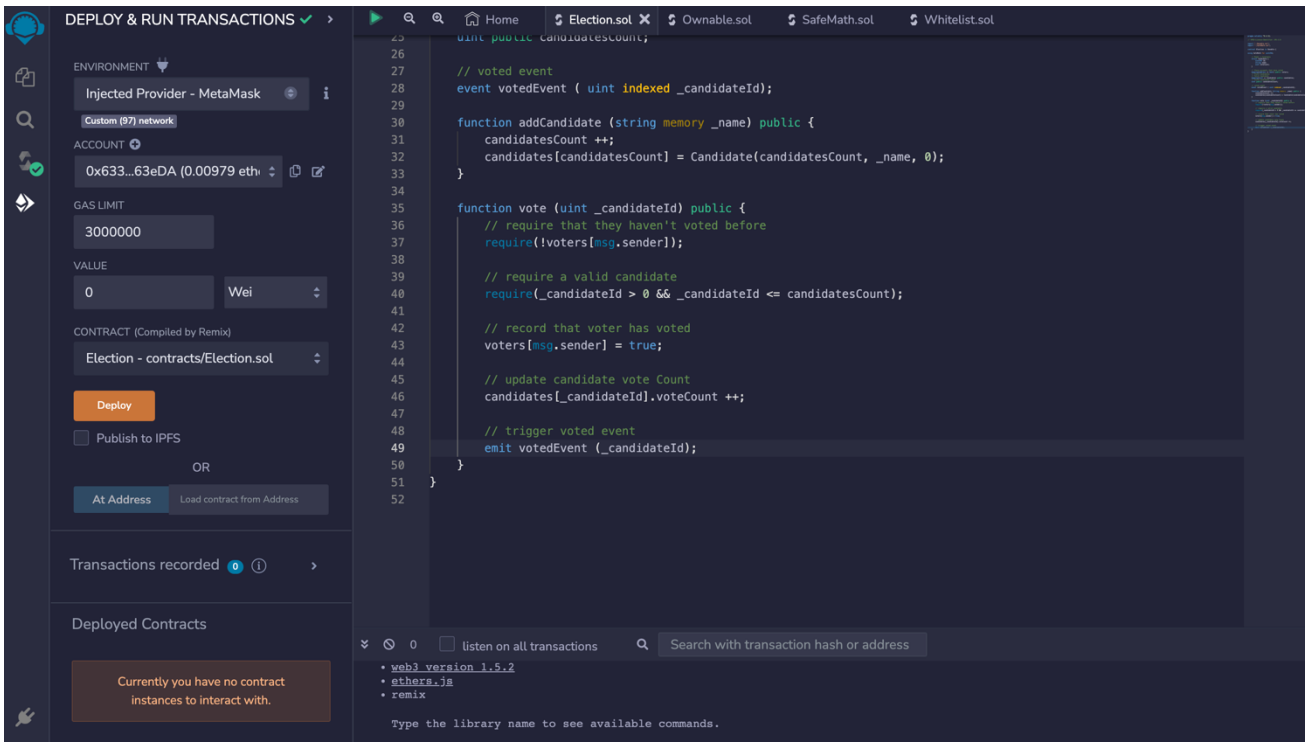
```
        {
                "anonymous": false,
                "inputs": [
                        {
                                "indexed": true,
                                "name": "_candidateId",
                                "type": "uint256"
                        }
               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

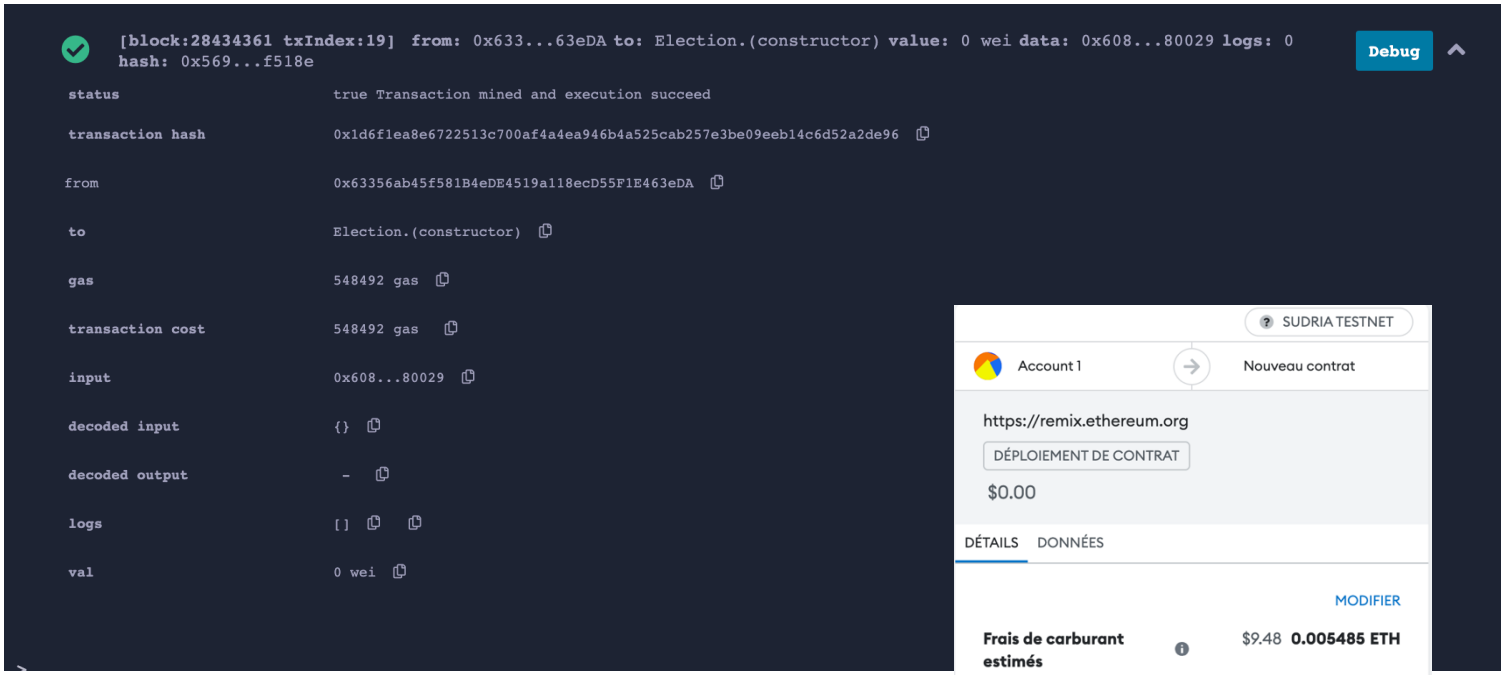

Bytecode :

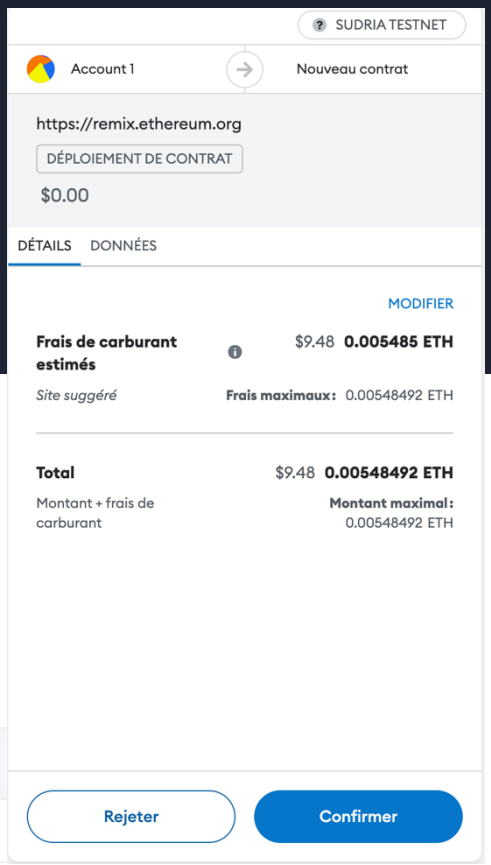

```
608060405233600080610100a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffff
ffffffffffffffffffffffff160217905550610897806100536000396000f3006080604052600436106100835760
00357c0100000000000000000000000000000000000000000000000000000000900463ffffffff168063
0121b93f146100885780632d35a8a2146100b55780633477ee2e146100e0578063462e91ec14610194
5780638da5cb5b146101fd578063a3ec138d14610254578063f2fde38b146102af575b600080fd5b348
015610094576000080fd5b506100b360048036038101908080359060200190929190505050506102f2565
b005b3480156100c157600080fd5b506100ca610415565b604051808281526020019150506040518
0910390f35b3480156100ec57600080fd5b5061010b600480360381019080803590602001909291905
0505061041b565b604051808481526020018060200183815260200182810382528481815181526020
0191508051906020019080838360005b838110156101575780820151818401526020810190506101 3c
565b505050509050908101906601f168015610184578082038051600183602003610100 0a031916815
260200191505b509450505050506040518091039 0f35b3480156101a057600080fd5b506101fb60048
036038101908080359060200190820180359060200190808060 1f0160208091040260200160405190
810160405280939291908181526020018383808284378201915050505050509192919290505050610
4dd565b005b34801561020957600080fd5b5061021261261055a565b604051808273ffffffffffffffffffffffffff
ffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001915050604051809103 90f35b348015610
26057600080fd5b506102956004803603810190808035 73ffffffffffffffffffffffffffffffffffffffff1690602001
909291905050506105 7f565b604051808215151515815260200191505060405180910390f35b348015
```

6102bb57600080fd5b506102f06004803603810190808035737fffffffffffffffffffffffffffffffffffffff1690602
0019092919050505061059f565b005b600160003373ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffff
ffffffffffffffffffffffff16815260200190815260200160002060009054906101000a900460ff161515156103
4b57600080fd5b60008111801561035d57506003548111155b1515610368576000 80fd5b600180600
03373ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152602001
60002060006101000a81548160ff0219169083151502179055506002600082815260200190815260200
160002060020160008154809291906001019190505550807ffff3c900d938d21d0990d786e819f29b8
d05c1ef587b462b939609625b684b16604051604051809103 90a250565b60035481565b6002602052
80600052604060002060009150905080600001549080600101805460018160011615610100020316 6
002900480601f0160208091040260200160405190810160405280929190818152602001828054600 1
816001161561010002031660029004801561 04cd5780601f106104a257610100808354040283529 16
02001916104cd565b820191906000526020600020905b81548152906001019060200180831161 04b0
57829003601f168201915b505050505090806002015490508 3565b600360008154809291906001019
190505550606060405190810160405280600354815260200182815260200160008152506002600060
0354815260200190815260200160002060008201518160000155602082015181600101908051906 02
0019061054992919061 07c6565b506040820151816002015590505050565b600080905490610 1000a
900473ffffffffffffffffffffffffffffffffffffffff1681565b600160205280600052604060002060009150549061 0
1000a900460ff1681565b600080905490610 1000a900473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffffffffff163373ffffffffffffffffffffffffffffffffffffffff161415156106 63576040517f08c379a000
0000000000000000000000000000000000000000000000000000000081526004018080602001828103825260198152602001807f4e6f7420617574686f72697a6564206f7065726174696f6e00000000000000
008152506020019150506040518091039 0fd5b600073ffffffffffffffffffffffffffffffffffffffff168173ffffffffffff
ffffffffffffffffffffffff161415151561070857 6040517f08c379a000000000000000000000000000000000
0000000000000000000000000815260040180806020018281038252601981526020018 07f41646472
6573732073686f756c646e2774206265207a65726f0000000000000081525060200191505060405180
910390fd5b8073ffffffffffffffffffffffffffffffffffffffff166000809054906101000a900473ffffffffffffffffffffffff
ffffffffffff1673ffffffffffffffffffffffffffffffffffffffff167f8be0079c531659141344cd1fd0a4f28419497f9722
a3daafe3b4186f6b6457e0604051604051809103 90a3806000806101000a81548173ffffffffffffffffffffff
ffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff160217905550 50565b82805460018160011
615610100020316600290049060005260206000209060 1f016020900481019282601f1061080757805
160ff191683800117855561083556 5b828001600101855582156108355791820 15b828111156108345
782518255916020019190600101906 10819565b5b5090506108429190610846565b5090565b610868
91905b8082111561086457600081600090555060010161084c565b5090565b905600a165627a7a723
05820c2b1daaf1ae81292340b32ac261cca45ecfd2e95a4ad3032438fcba078984ae80029

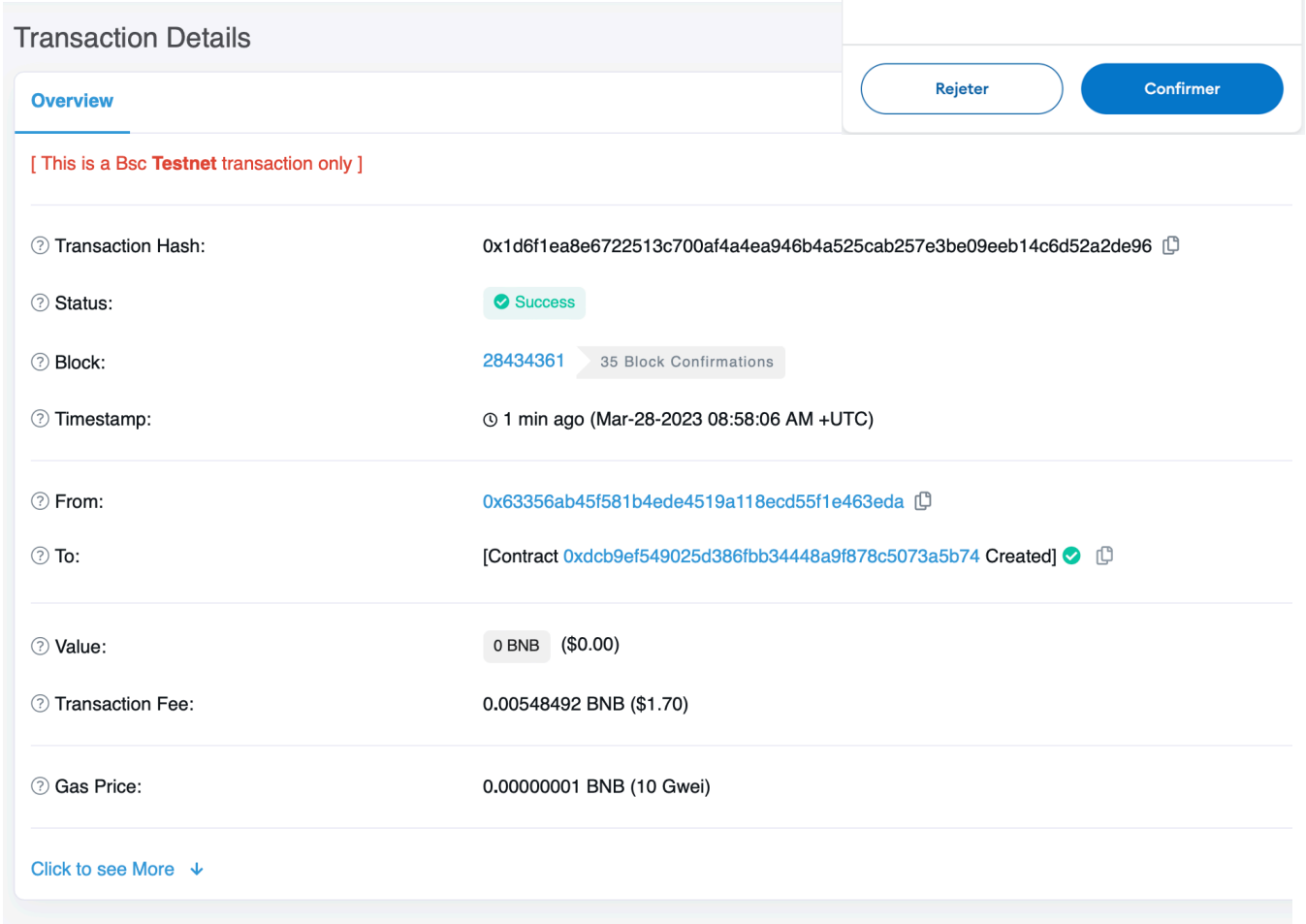
M) Déploiement sur smart contract « Election.sol » :



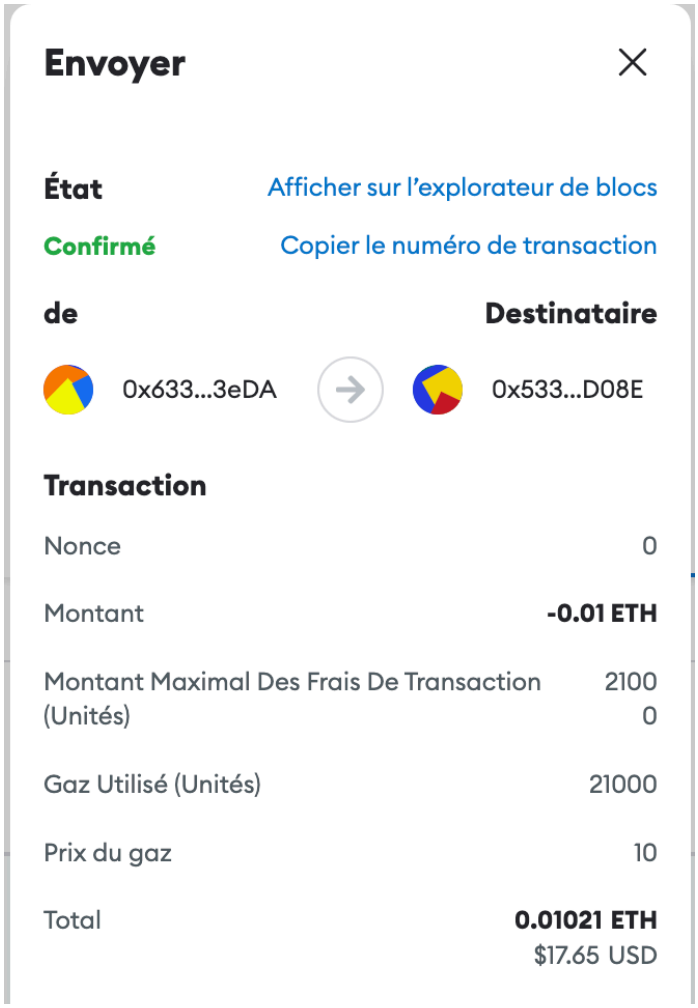




Adresse publique smart contract :
0xdcb9ef549025d386fbb34448a9f878c5073a5b74

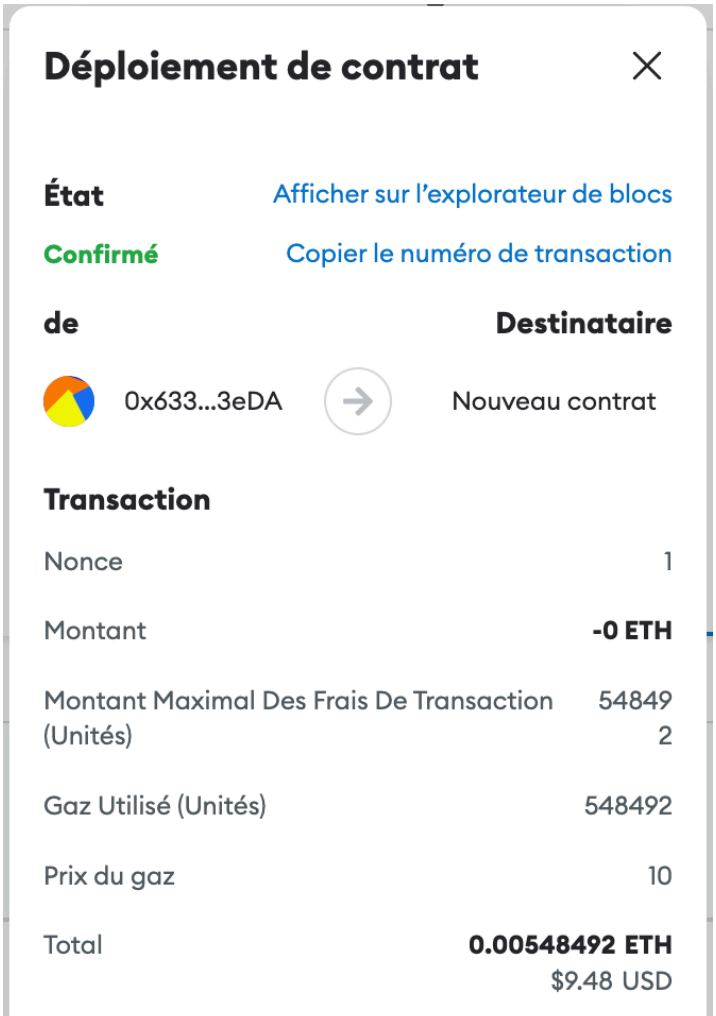


Les « Transactions fees » sont différents de la précédente transaction comme nous pouvons le voir ci-dessous :

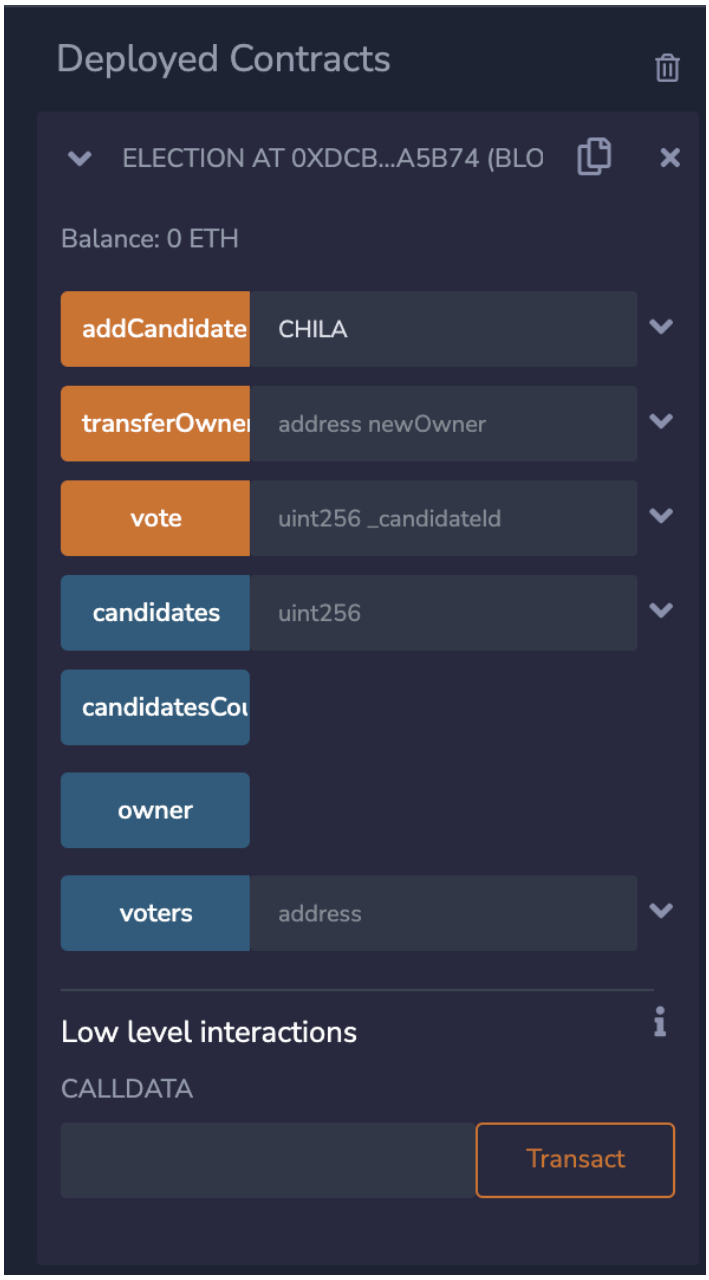


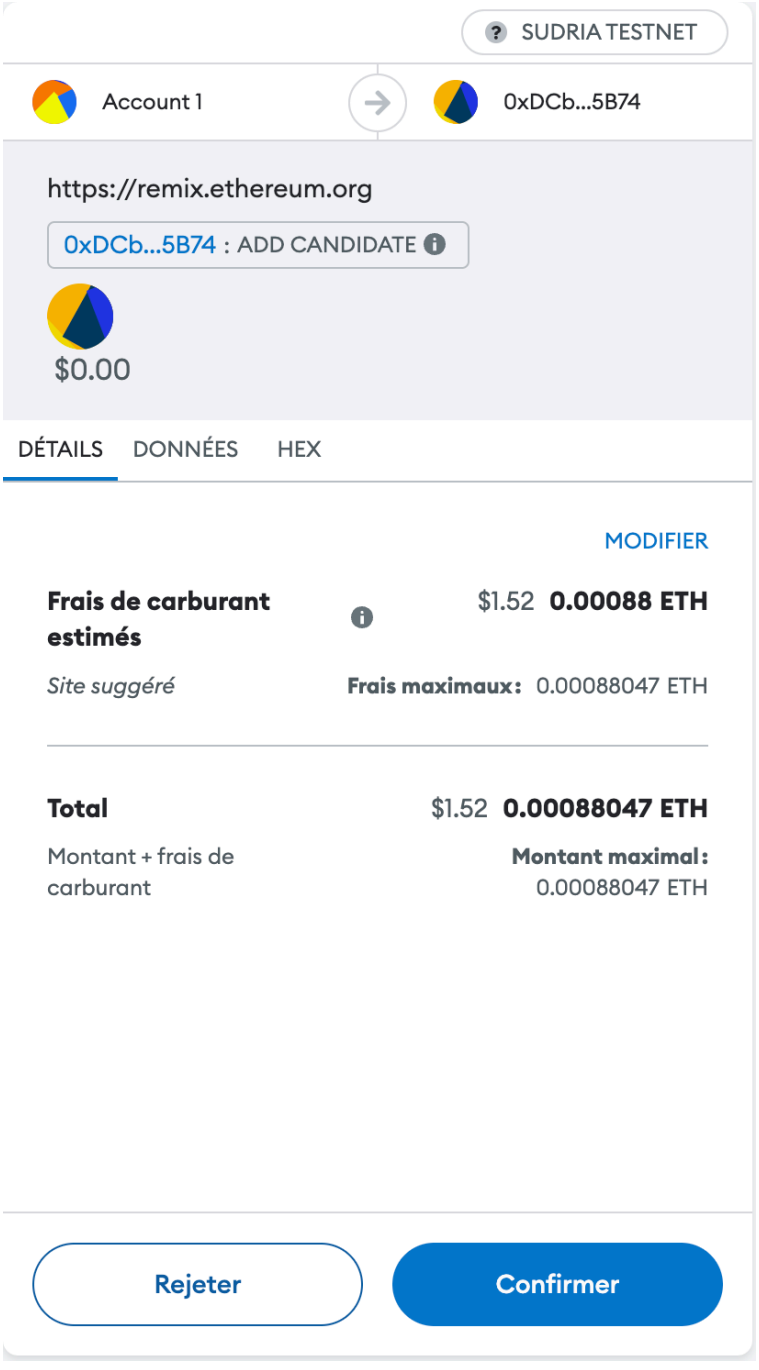
Montant maximal = 2100

Montant maximal = 54 849

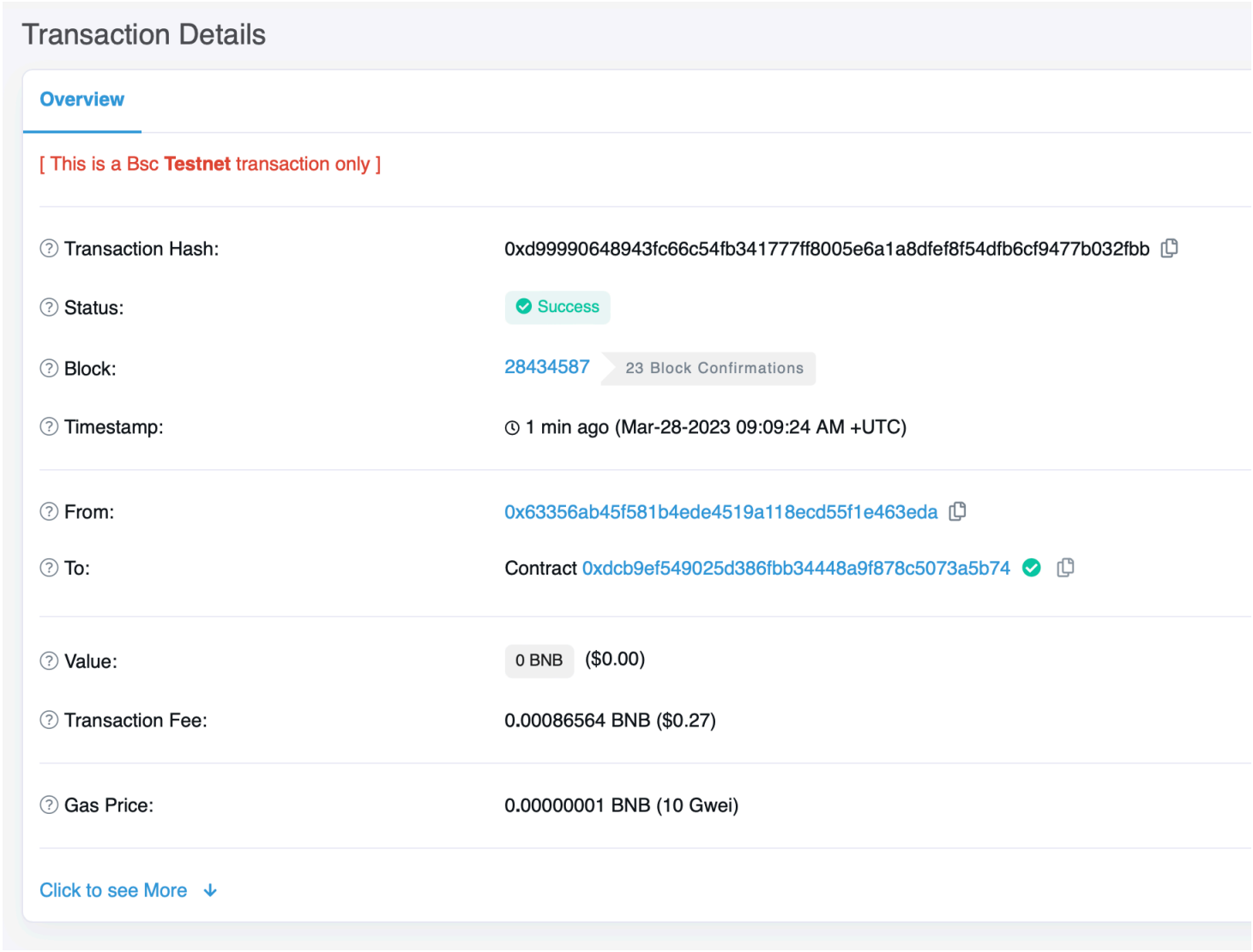


N) Ajout du premier candidat : CHILA

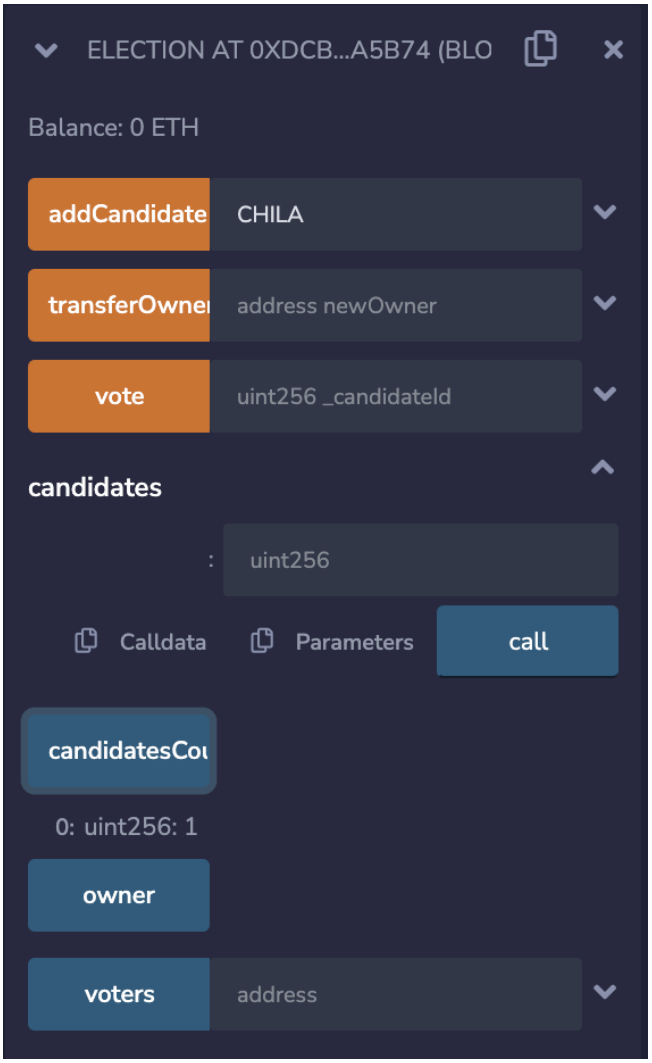




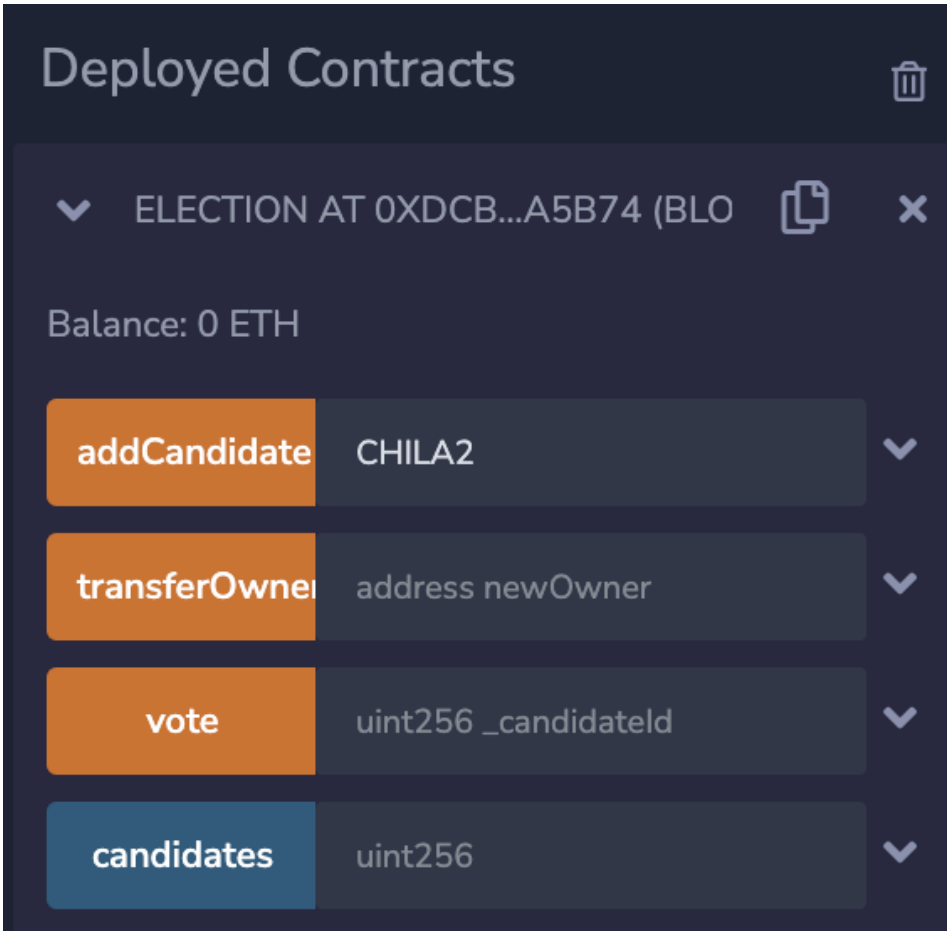
O) Détails de la transaction :

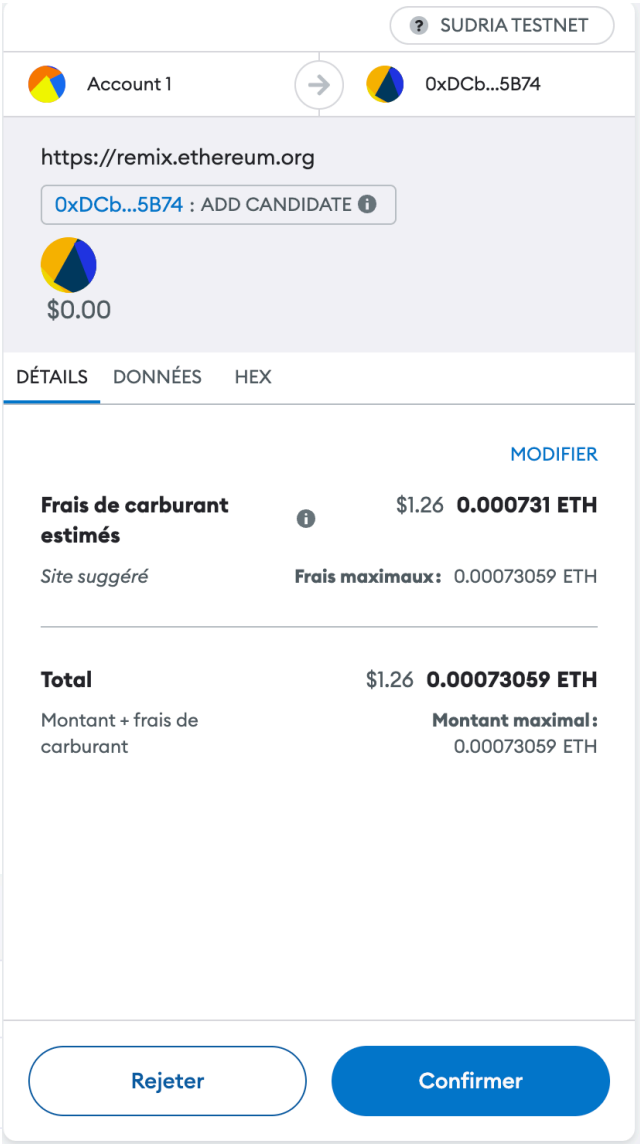


P) La valeur de CandidateID = 1



Q) Ajout du second candidat : CHILA2 et détail de la transaction :





Transaction Details

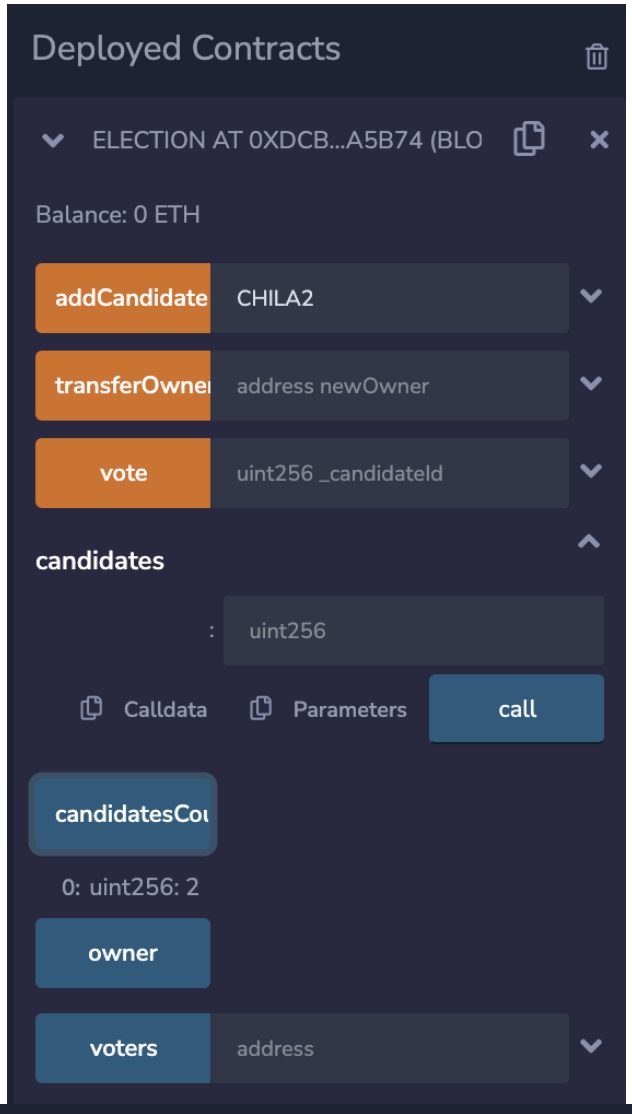
Overview

[This is a Bsc **Testnet** transaction only]

Transaction Hash:	0xb14b3fd5f1ce2c07027a007249670820f4aa2e9941c0a87c1b5e8cb6438f6d81
Status:	Success
Block:	28434884 20 Block Confirmations
Timestamp:	1 min ago (Mar-28-2023 09:24:15 AM +UTC)
From:	0x63356ab45f581b4ede4519a118ecd55f1e463eda
To:	Contract 0xdcb9ef549025d386fbb34448a9f878c5073a5b74
Value:	0 BNB ($0.00)
Transaction Fee:	0.00071576 BNB ($0.22)
Gas Price:	0.00000001 BNB (10 Gwei)

Click to see More

R) La valeur du second CandidateID = 2 :

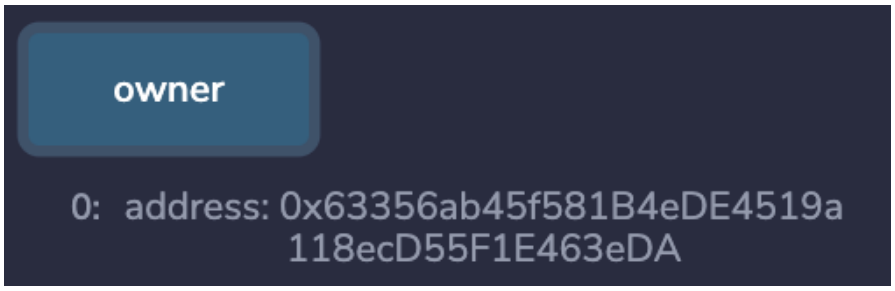



```
call to Election.candidatesCount

CALL  [call] from: 0x63356ab45f581B4eDE4519a118ecD55F1E463eDA to: Election.candidatesCount() data: 0x2d3...5a8a2

from                0x63356ab45f581B4eDE4519a118ecD55F1E463eDA
to                  Election.candidatesCount() 0xDCb9ef549025d386Fbb34448a9f878c5073A5B74
input               0x2d3...5a8a2
decoded input       {}
decoded output      {
                        "0": "uint256: 2"
                    }
logs                []
```

S) Adresse du propriétaire du contrat :

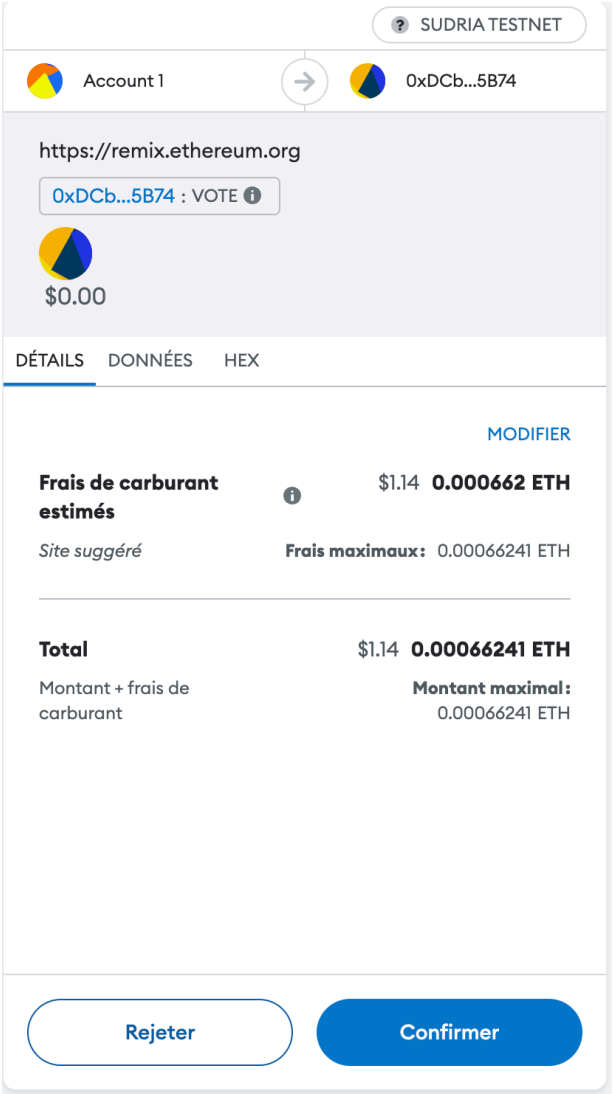



```
call to Election.owner

CALL  [call] from: 0x63356ab45f581B4eDE4519a118ecD55F1E463eDA to: Election.owner() data: 0x8da...5cb5b
```

T) Premier vote pour l'un des candidats (CHILA) et détails de la transaction :

CandidateID = 1 (CHILA)



Transaction Details

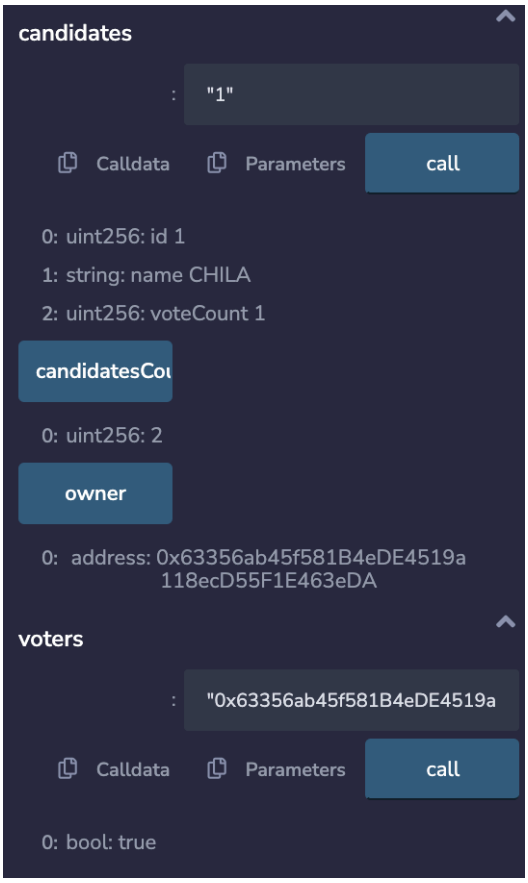
Overview | Logs (1)

[This is a Bsc **Testnet** transaction only]

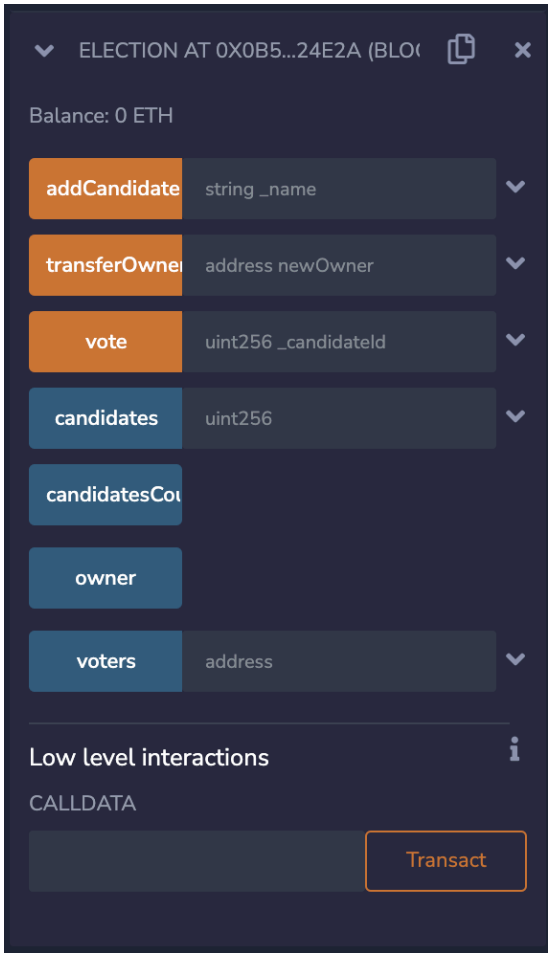
Transaction Hash:	0x0a657cae4f721477eb8829c11bf7eacfb5f2fda76610efd920046d9e139ff9fc
Status:	Success
Block:	28435051 7 Block Confirmations
Timestamp:	25 secs ago (Mar-28-2023 09:32:36 AM +UTC)
From:	0x63356ab45f581b4ede4519a118ecd55f1e463eda
To:	Contract 0xdcb9ef549025d386fbb34448a9f878c5073a5b74
Value:	0 BNB ($0.00)
Transaction Fee:	0.00066241 BNB ($0.21)
Gas Price:	0.00000001 BNB (10 Gwei)

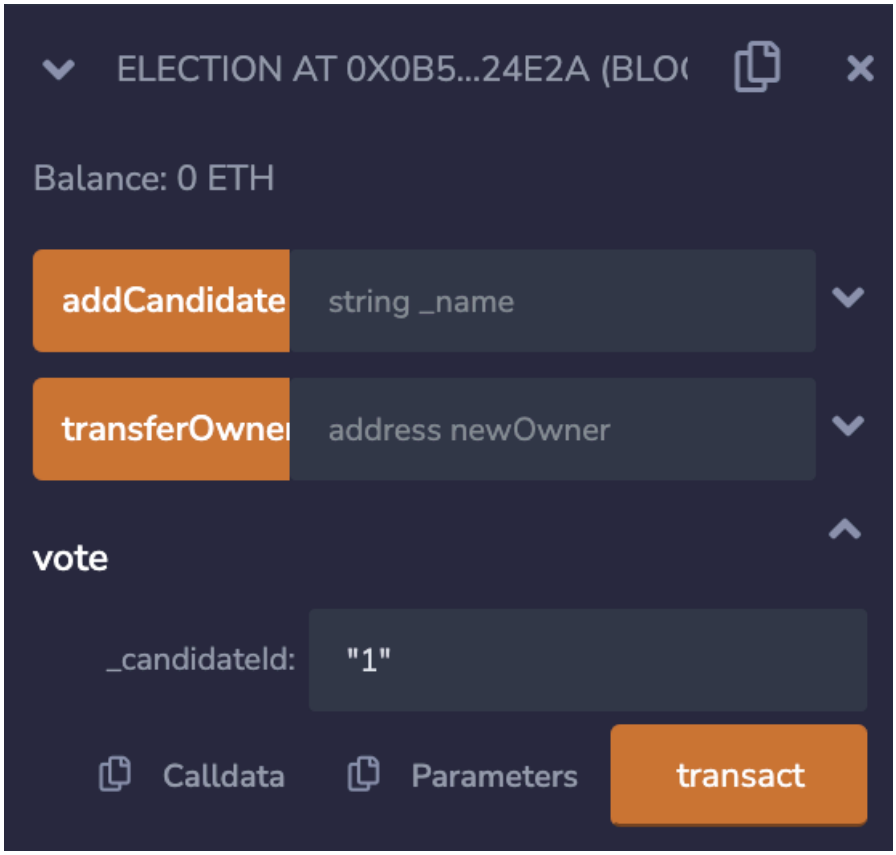
Click to see More

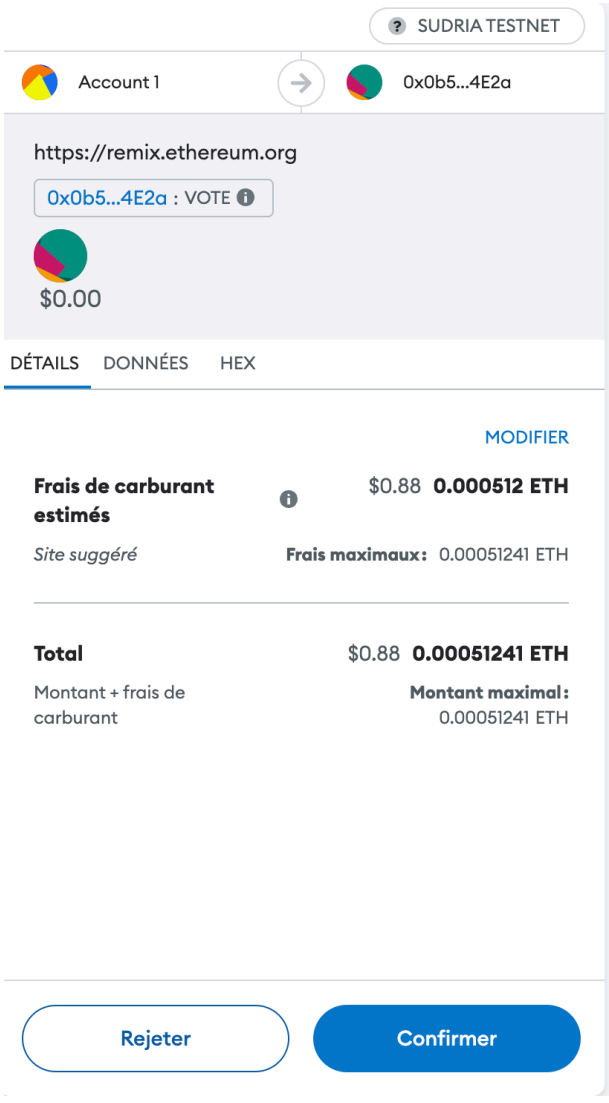
U) Vérification que le vote a bien été pris en compte : voteCount 1



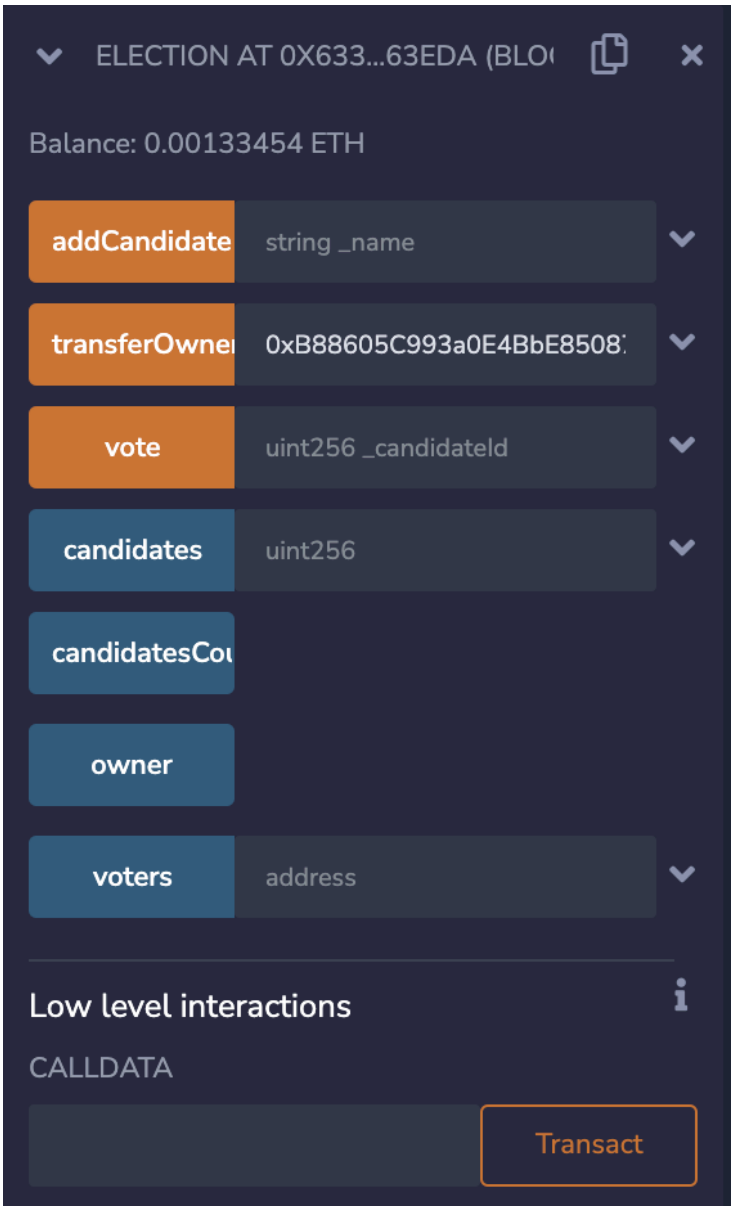
V) Interaction avec le contrat de Maxim TARAN :

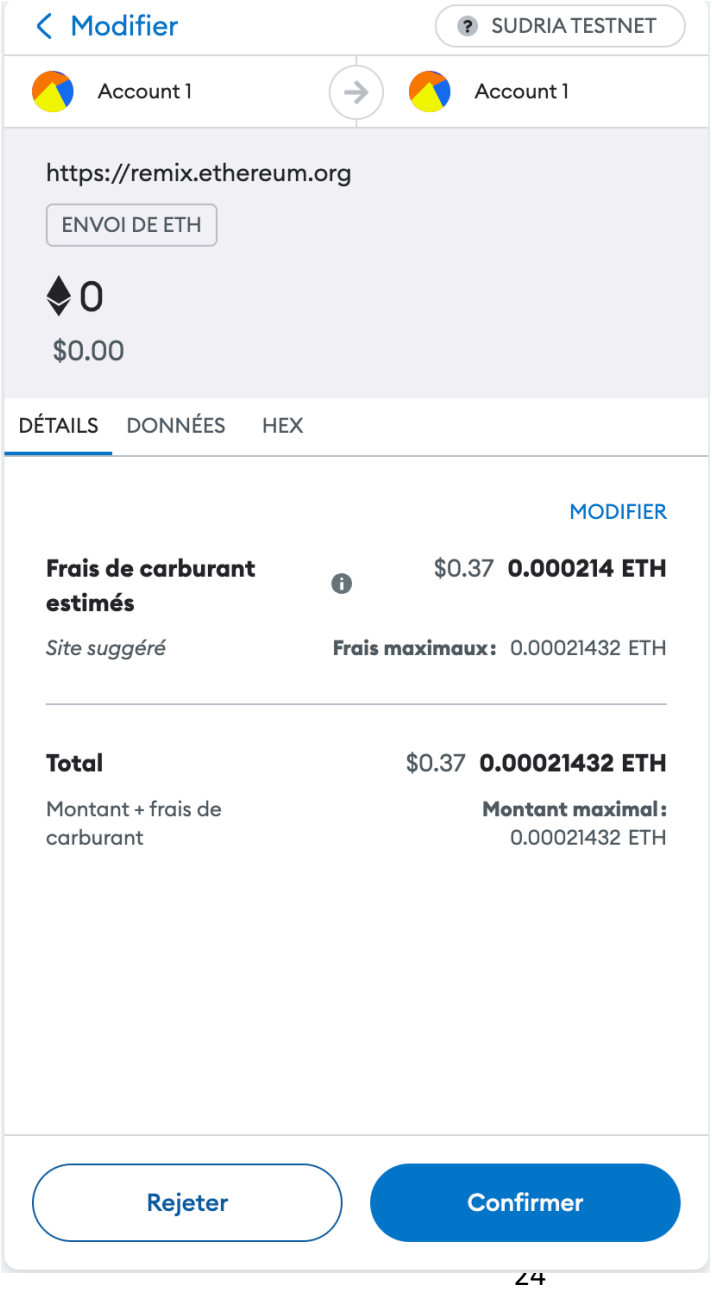


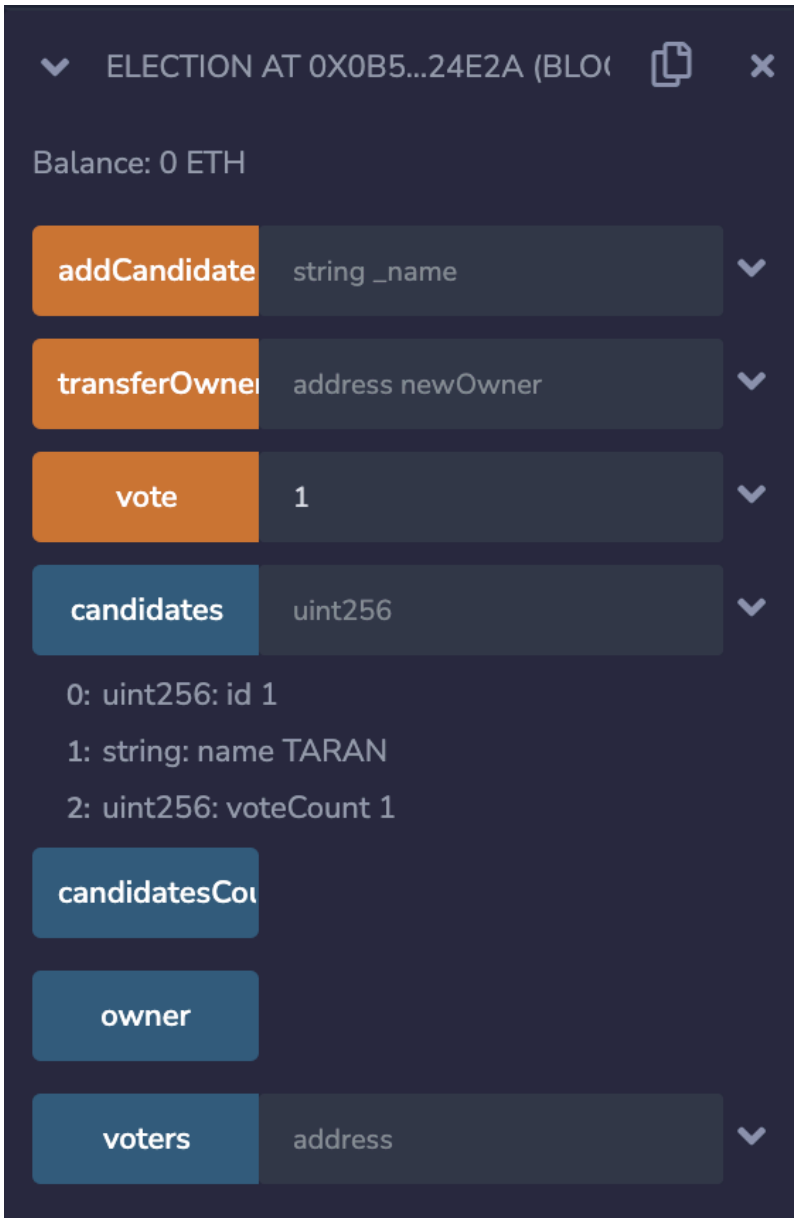




W) Transfert de la propriété de Maxim TARAN en lui demandant son adresse publique :
Adresse publique : 0xB88605C993a0E4BbE850879A3f2Ccf974Ad14E0B







Name TARAN

X) <u>Pourquoi sécuriser l'appel de la fonction addCandidate afin d'être le seul à pouvoir gérer les candidats ?</u>

La fonction addCandidate est une fonction qui permet d'ajouter un candidat à un système ou une application. La raison pour laquelle il est important de sécuriser l'appel à cette fonction est liée à la gestion des privilèges et des autorisations.Si l'appel de la fonction addCandidate n'est pas sécurisé, cela signifie que n'importe qui peut ajouter des candidats à la base de données ou au système. Cela peut potentiellement causer des problèmes de sécurité, car des personnes malveillantes pourraient ajouter des candidats fictifs ou corrompus, ou modifier les informations des candidats existants.

En sécurisant l'appel à la fonction addCandidate, on s'assure que seules les personnes ayant les autorisations nécessaires peuvent ajouter des candidats. Cela permet de limiter les risques de fraude ou de compromission des données.

Y) Modification du code afin que je sois le seul à pouvoir ajouter un nouveau candidat :

→ Utilisation de *public onlyOwner*

```
30
31      function addCandidate (string memory _name) public onlyOwner {
32          candidatesCount ++;
33          candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
34      }
35
```