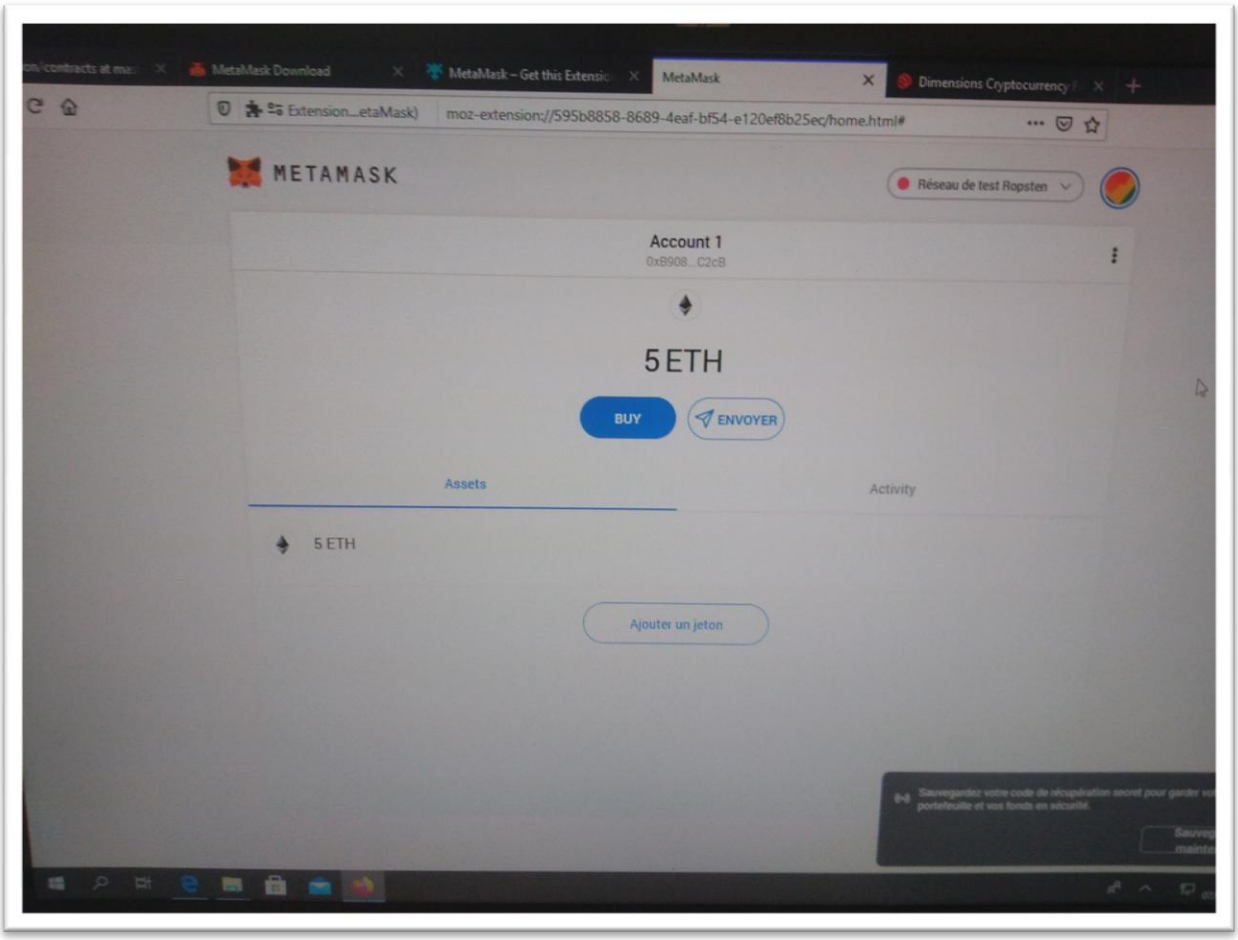
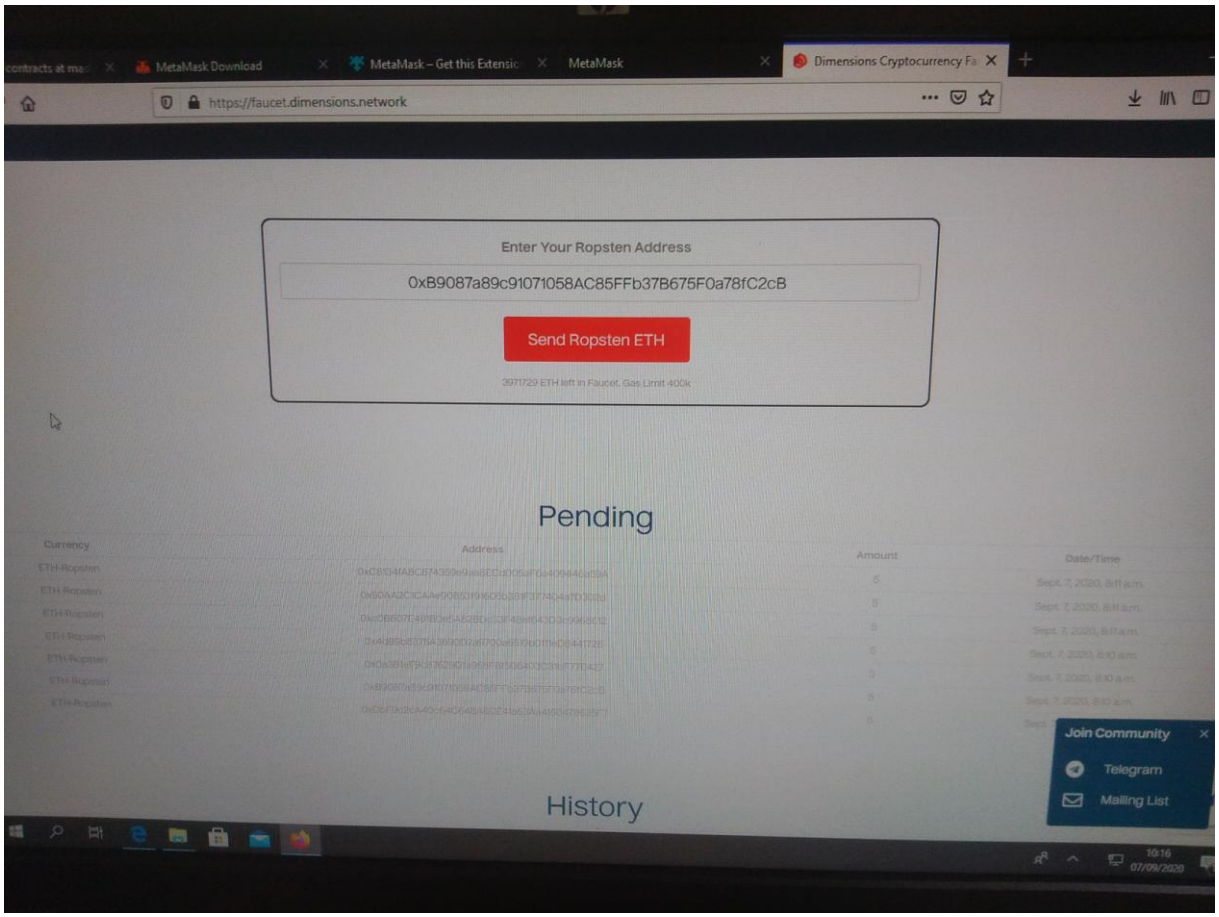
TRAVAUX PRATIQUE :
Développer, Deloyer et Interagir avec un contrat intelligent sur Ethereum

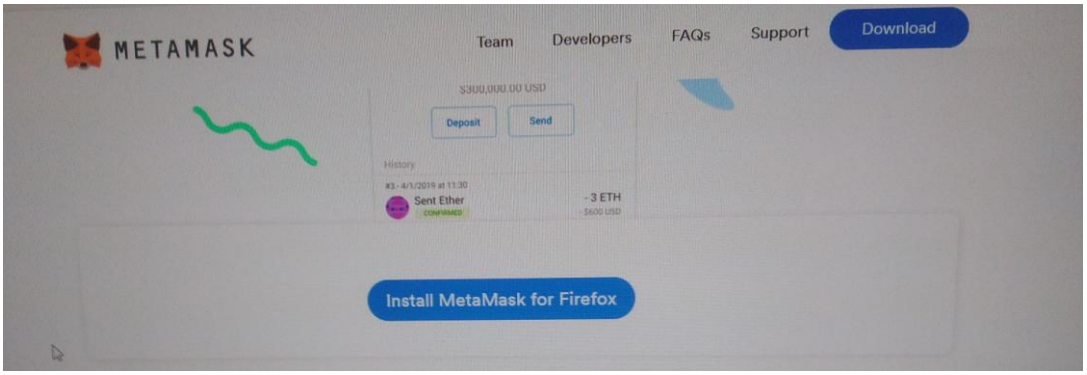
Première acquisition d'Ethers au travers de Réseau de test Ropsten



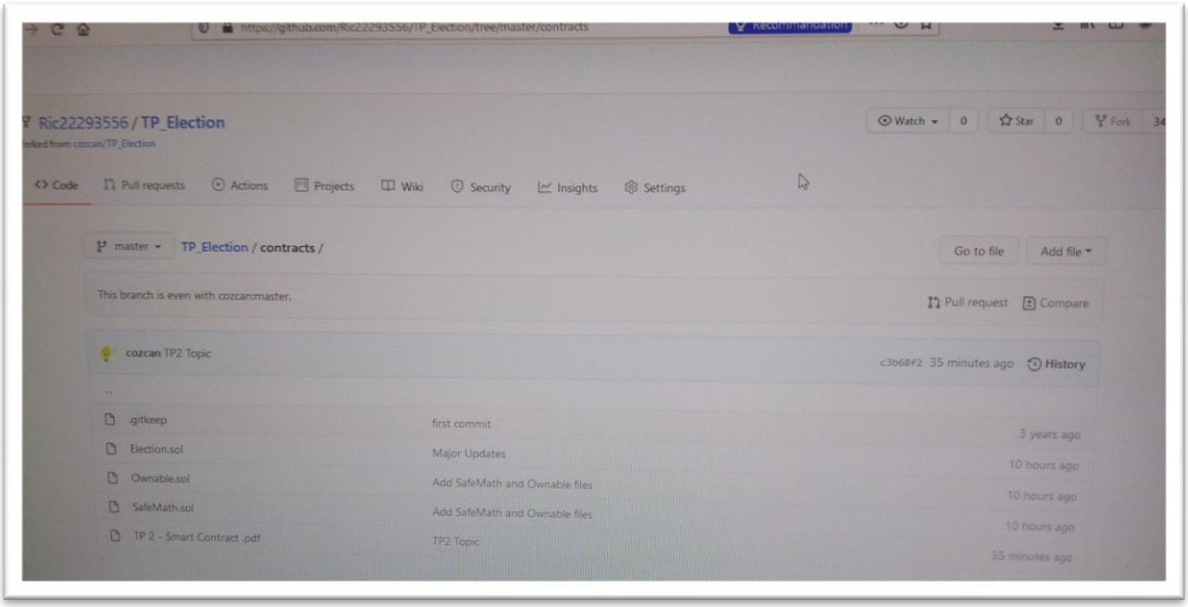
On peut alors observer le statu pending de la transaction, c'est-à-dire qu'elle est en attente d'exécution. Une fois la monnaie obtenue, celle-ci apparait dans le portefeuille.



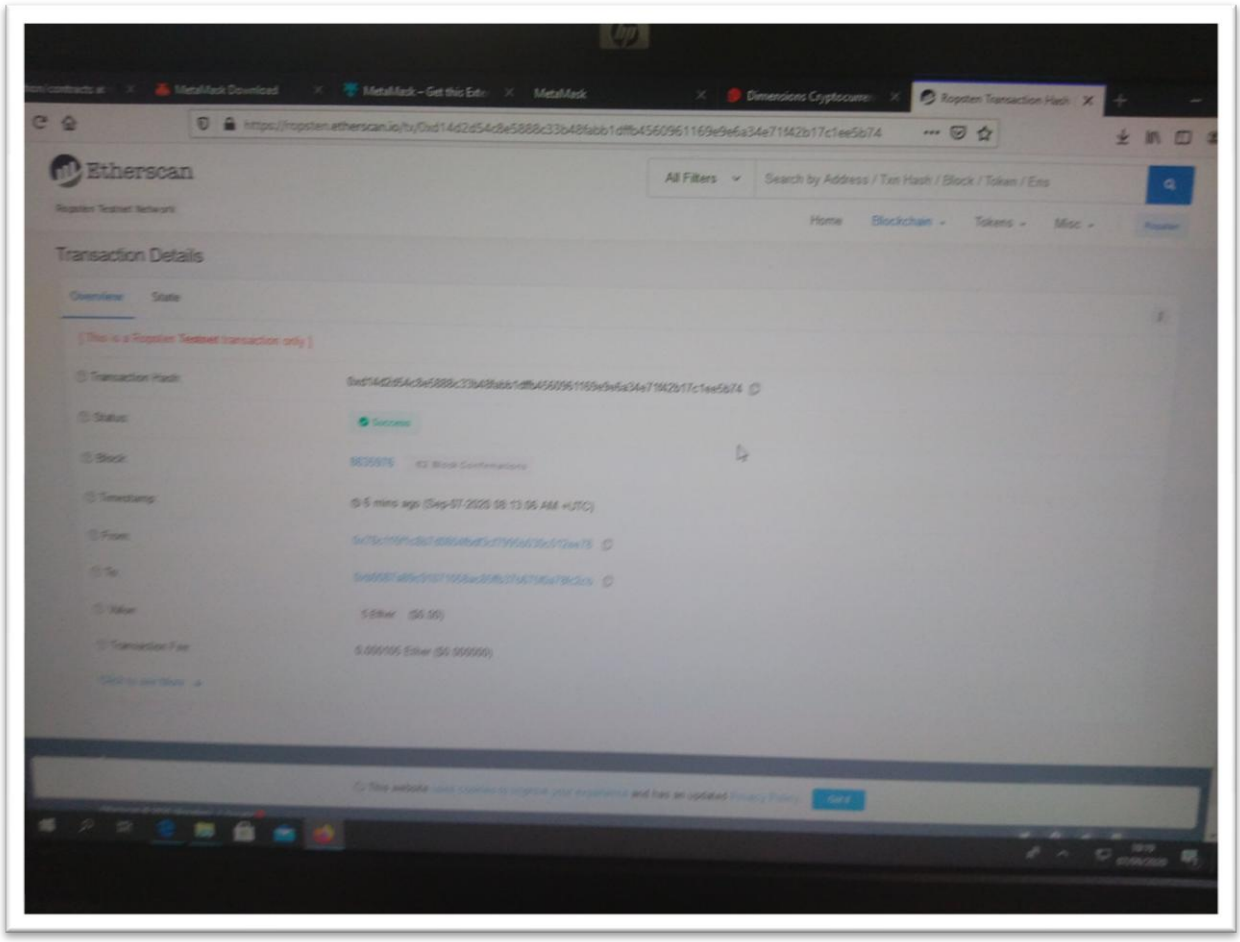
On installe alors Metamask pour firefox :

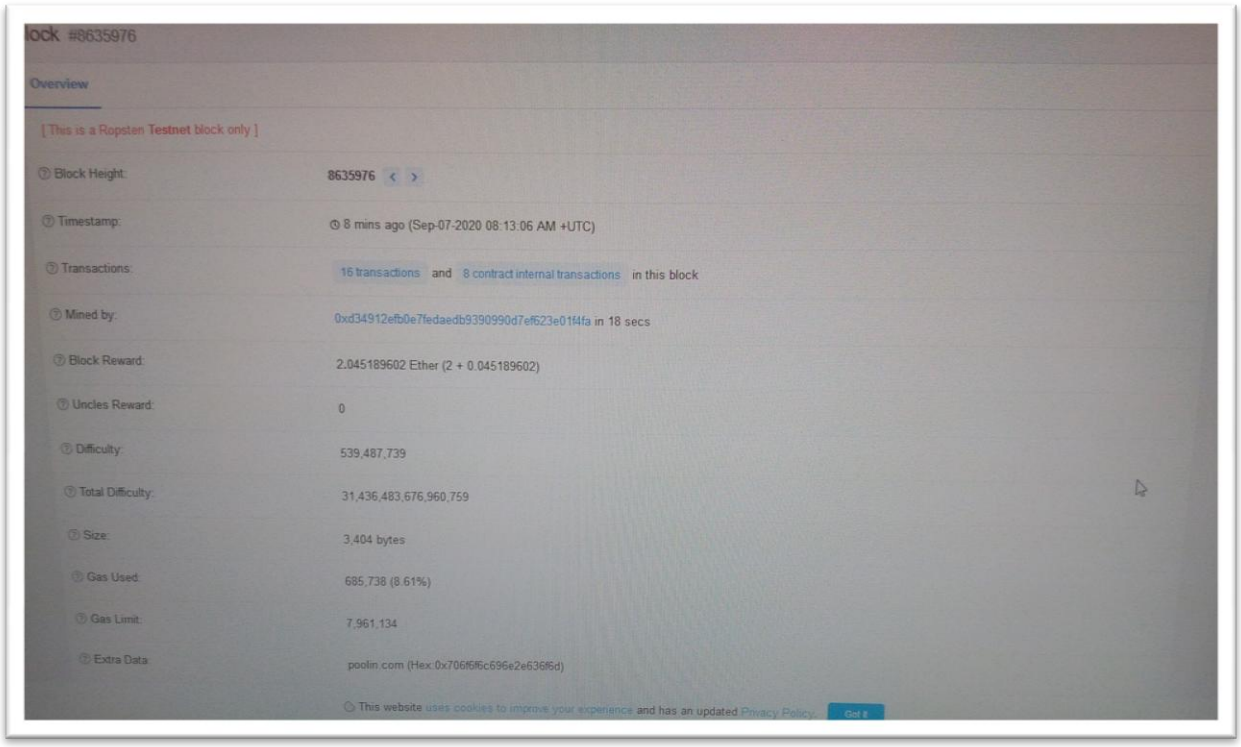


Une fois effectué, on peut alors charger les dossiers du tp dans notre compte personnel github :

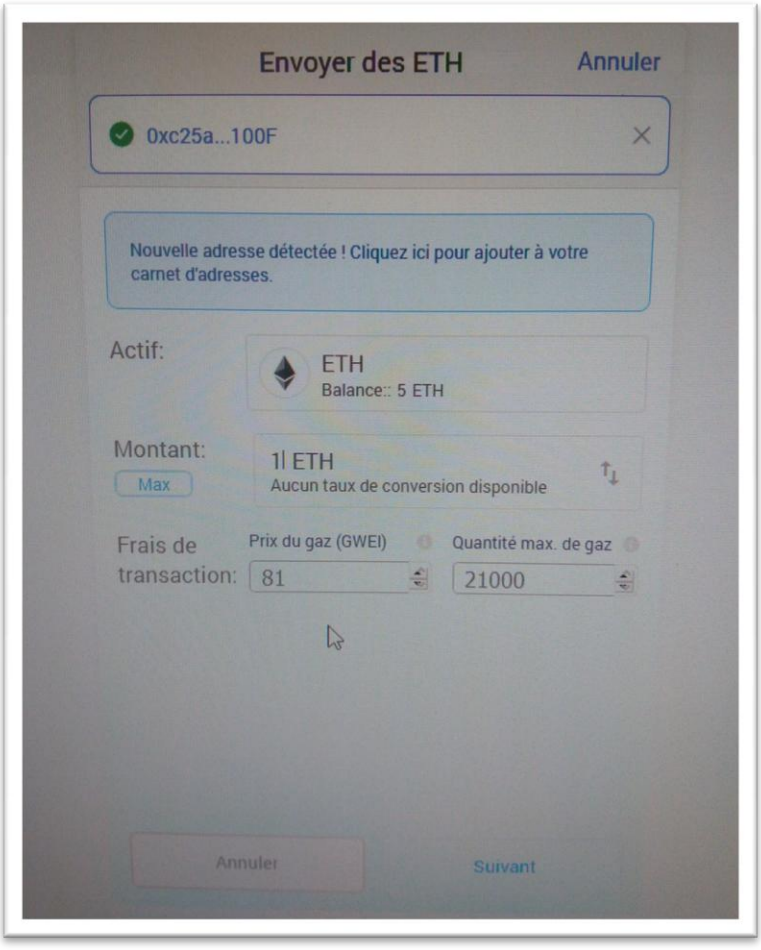


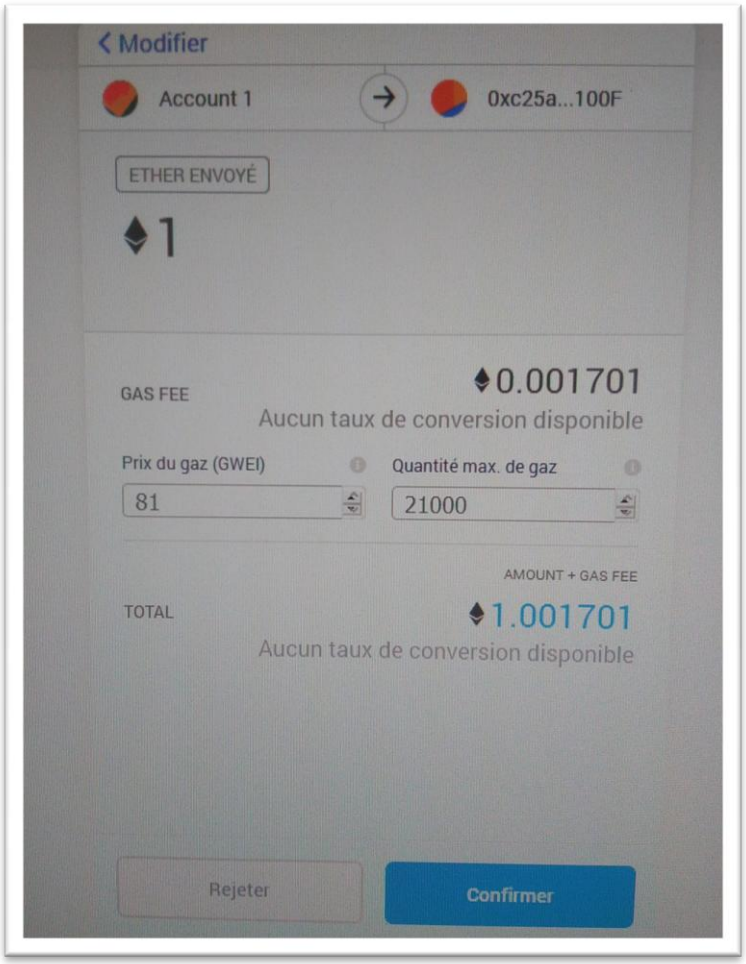
On peut aussi voir le detail de la transaction à l'aide d'Etherscan :



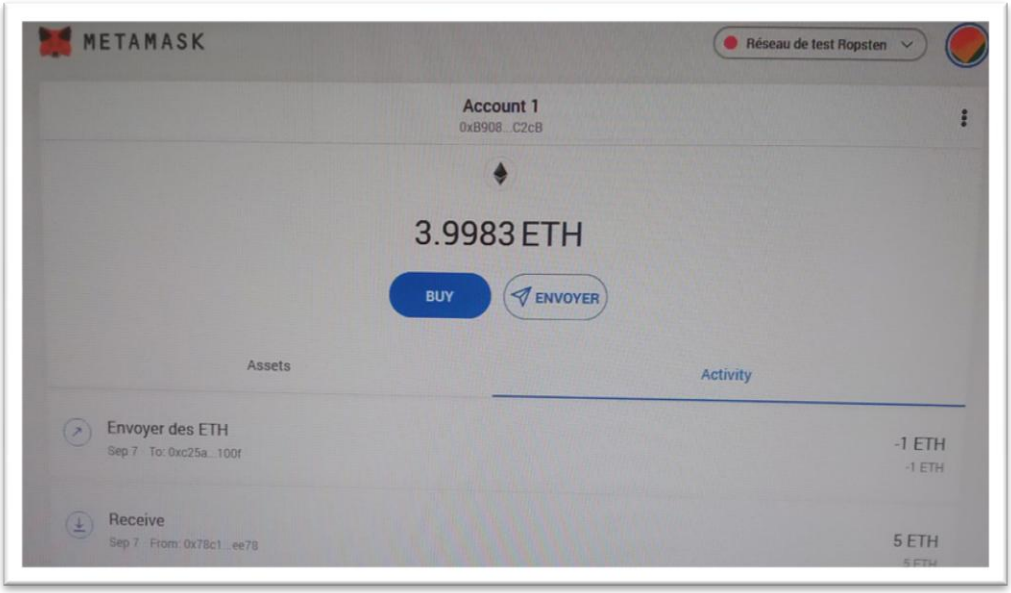


On peut alors envoyer des Ethers à une autre personne grâce à sa clé publique :



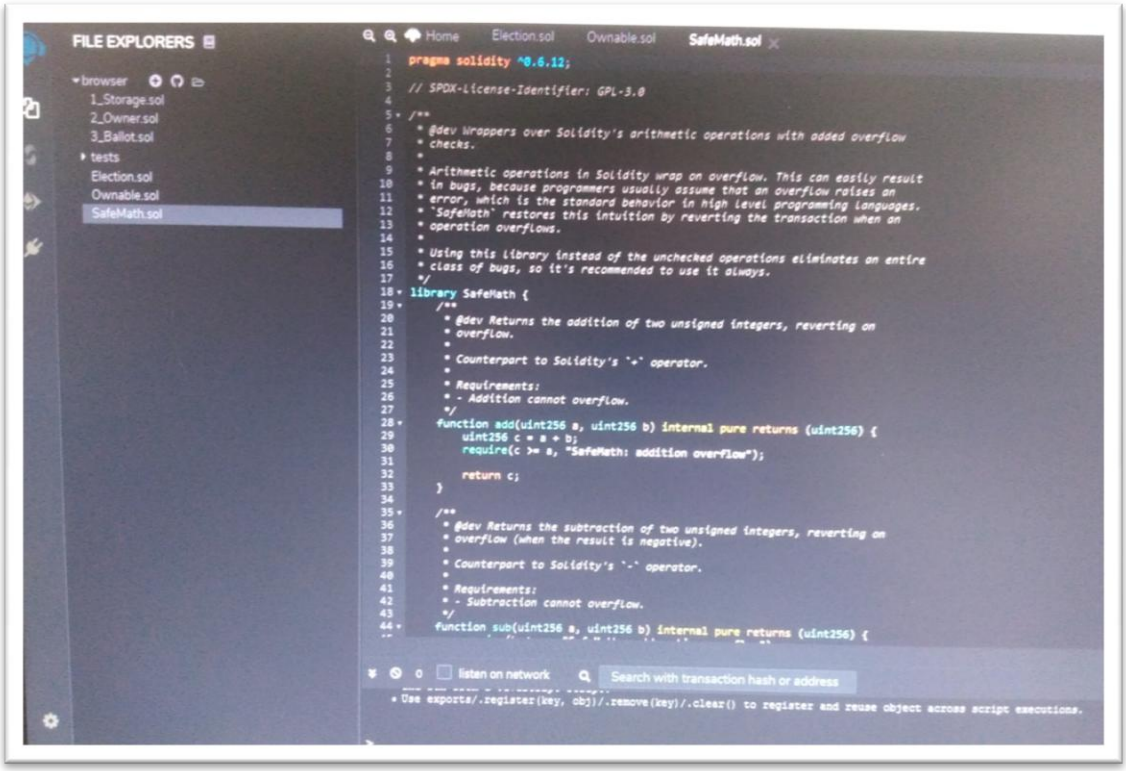


Après avoir envoyé les Ethers, forcement il y en a moins dans le portefeuille :

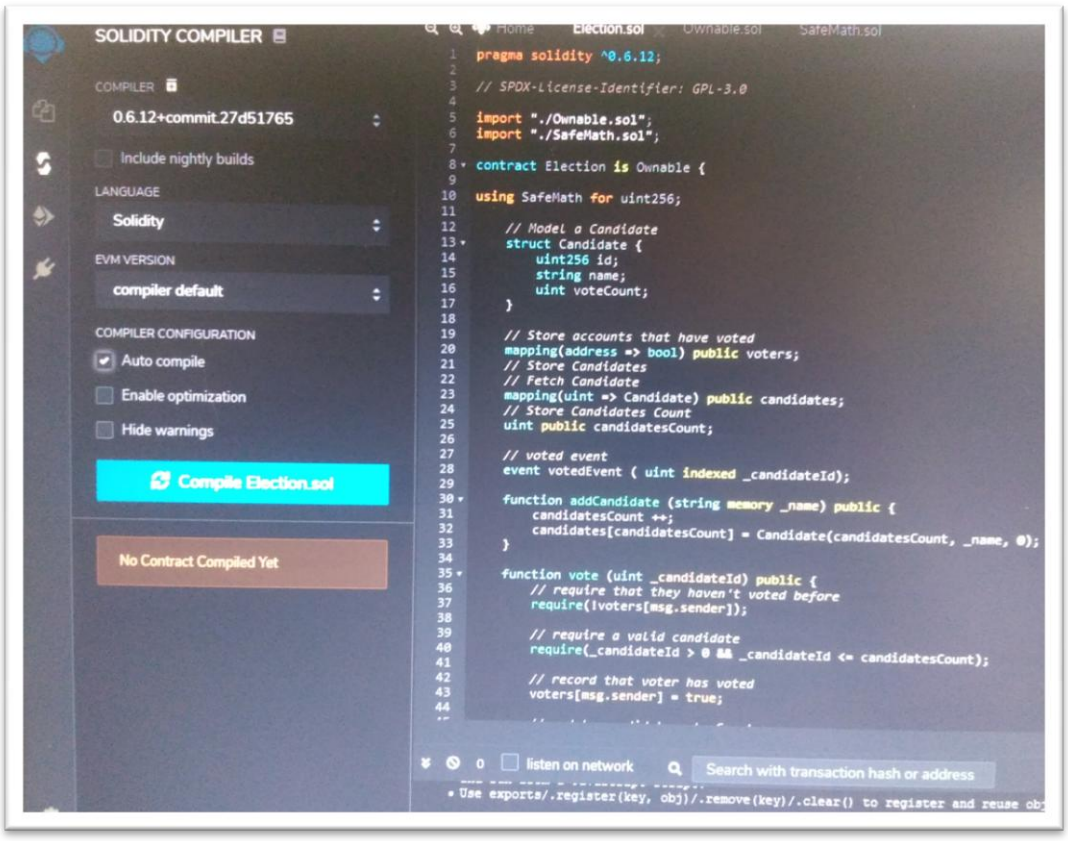


Les frais transactions diffèrent en fonction du nombre de transactions de users sur le réseau. Plus il y à de monde, plus il faut payer pour voir sa transaction s'effectuer dans les plus brefs délais. Cependant, il est difficile d'anticiper la congestion dans le réseau en mesurant les frais de gaz.

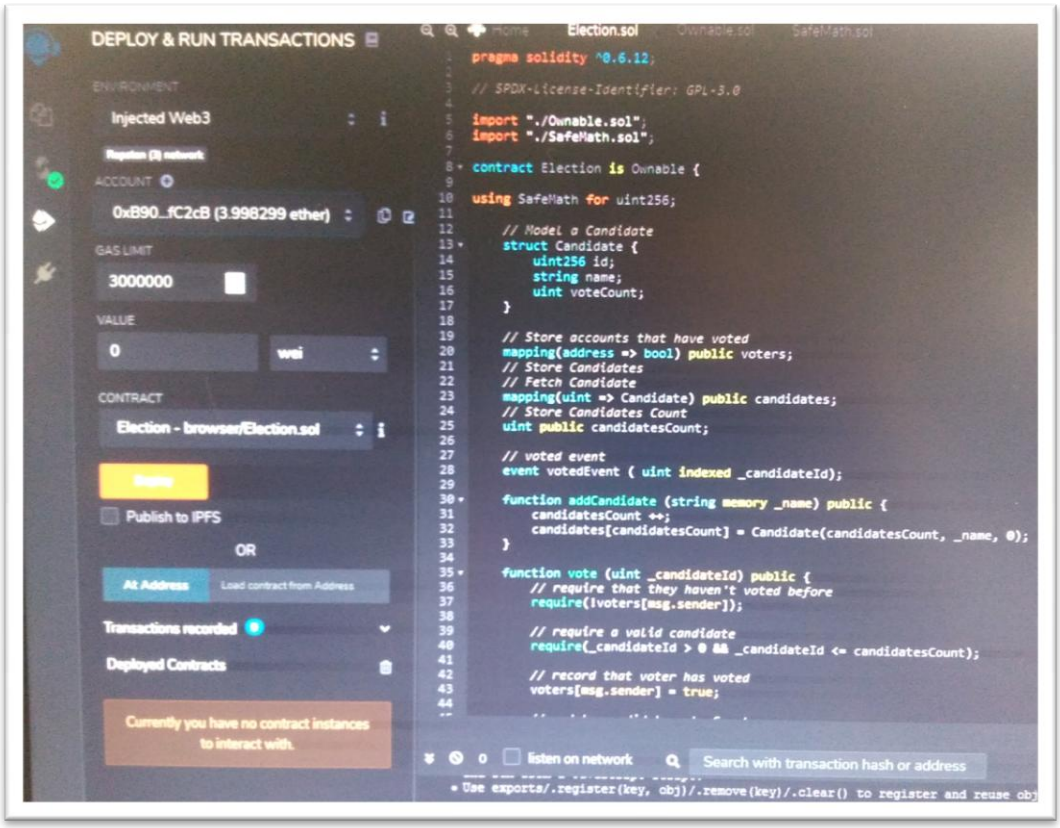
On peut alors se connecter sur Remix pour visualiser le code solidity d'un contrat :



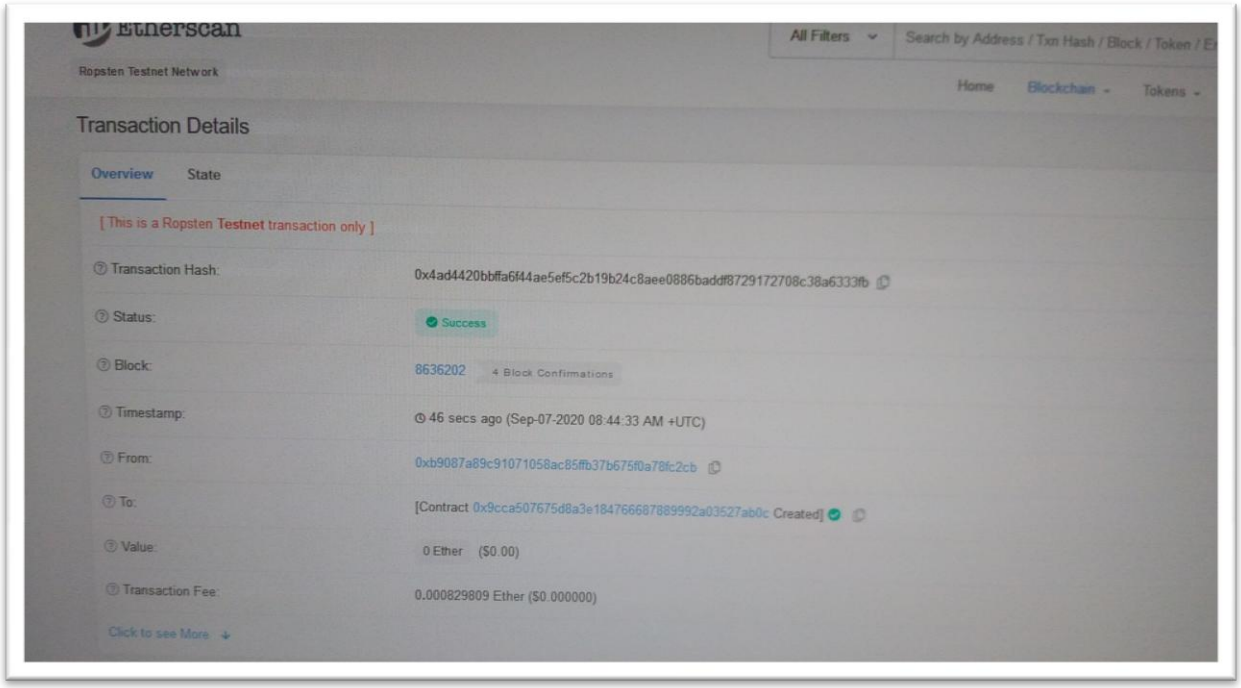
Une fois ajouté, le code doit être compilé :

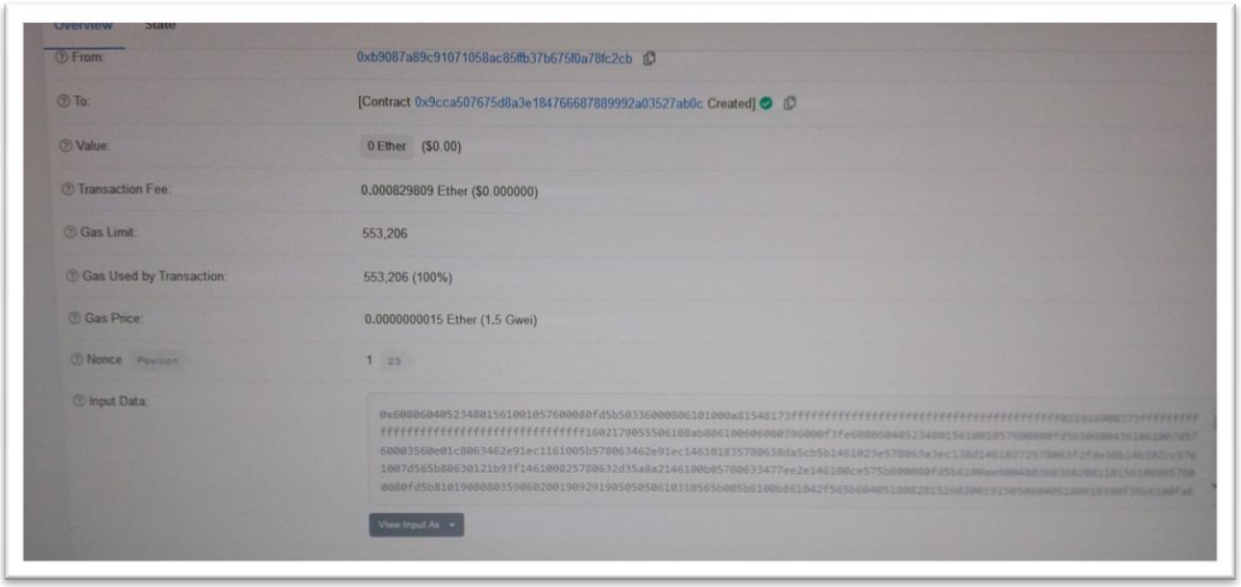


Enfin, il doit être distribué sur le réseau :

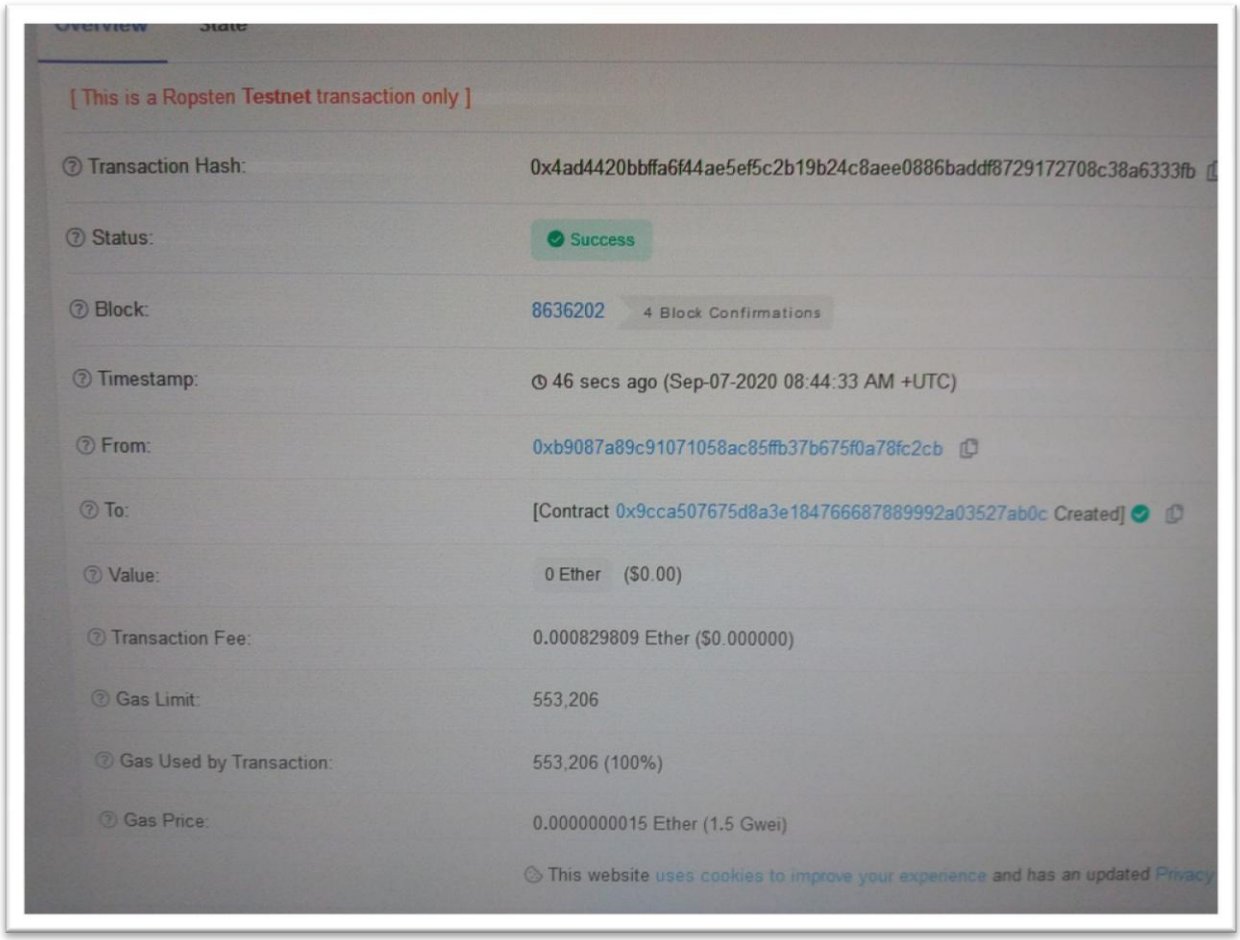


On peut alors vérifier la bonne distribution du code :

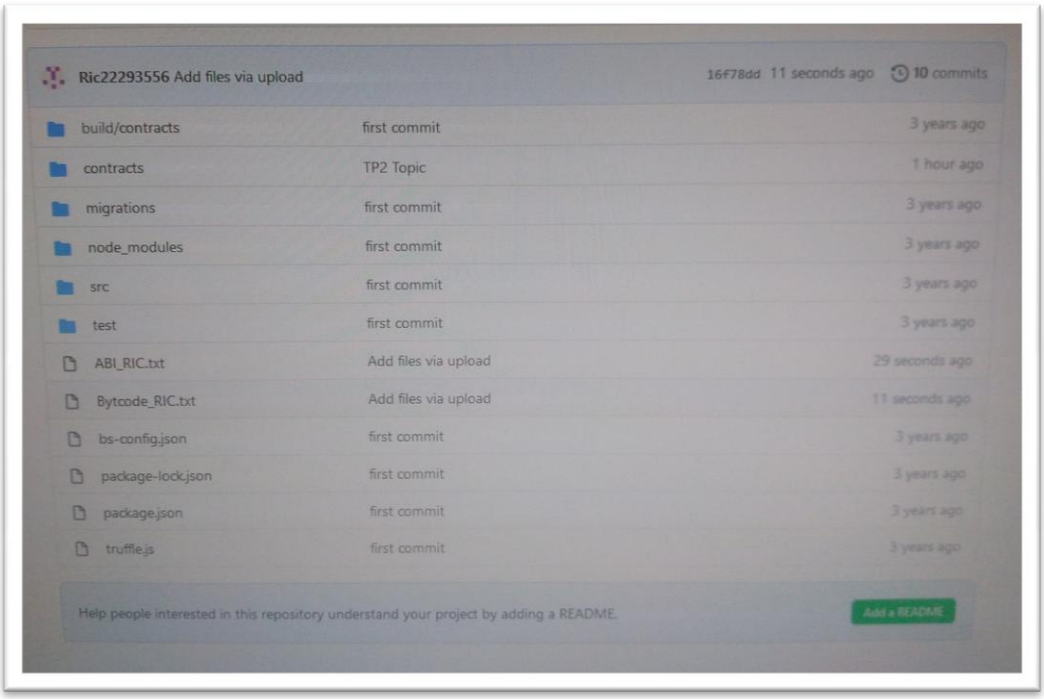




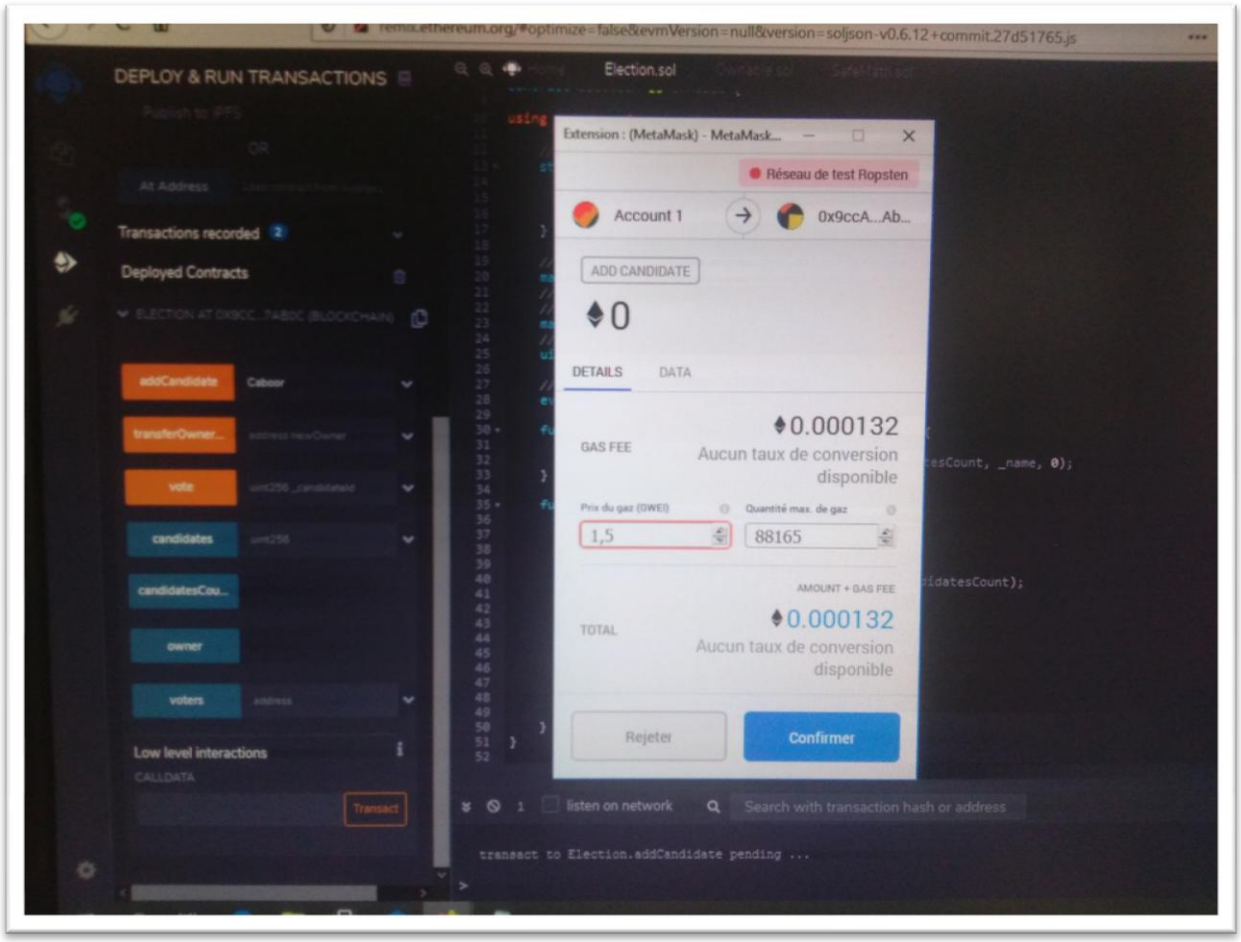
On retrouve notamment l'adresse du contrat dans le résumé de transaction, ici il s'agit de 0x9cca507675d8a3e184766687889992a03527ab0c :



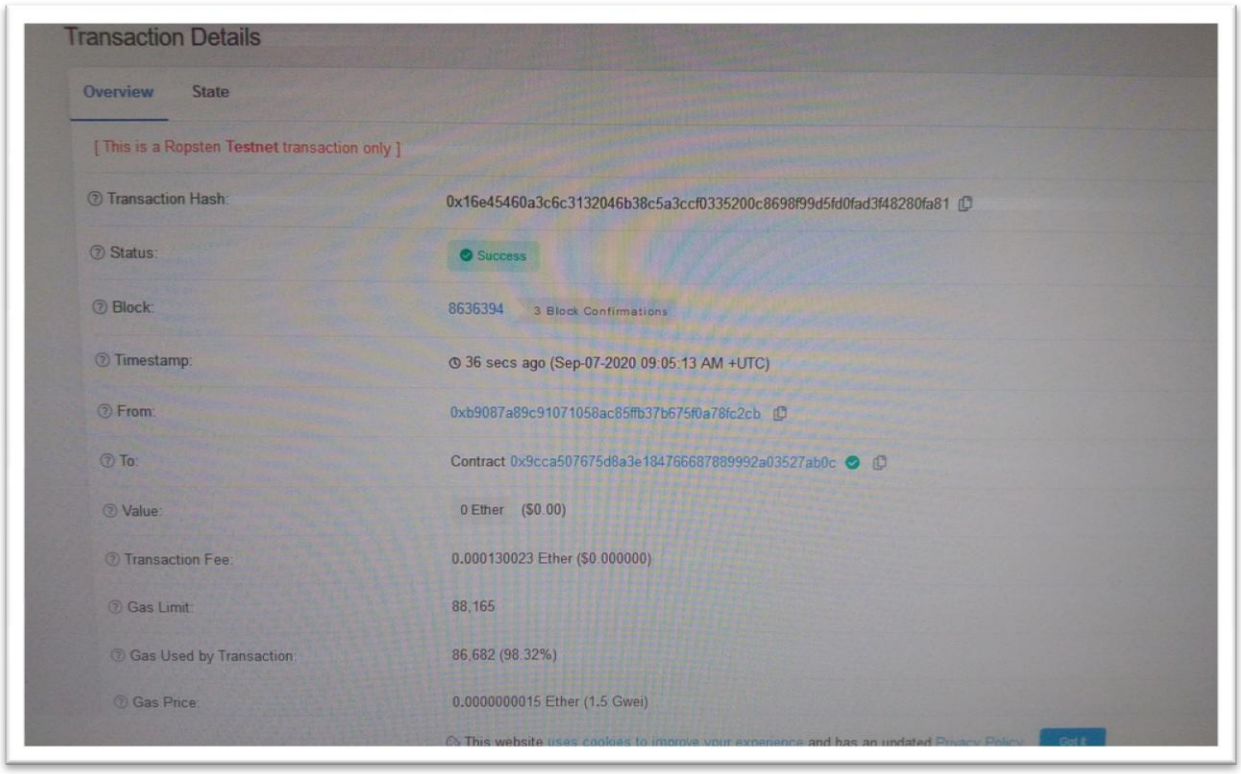
Une fois distribué, on peut charger le Bytes code et l'ABI dans github :

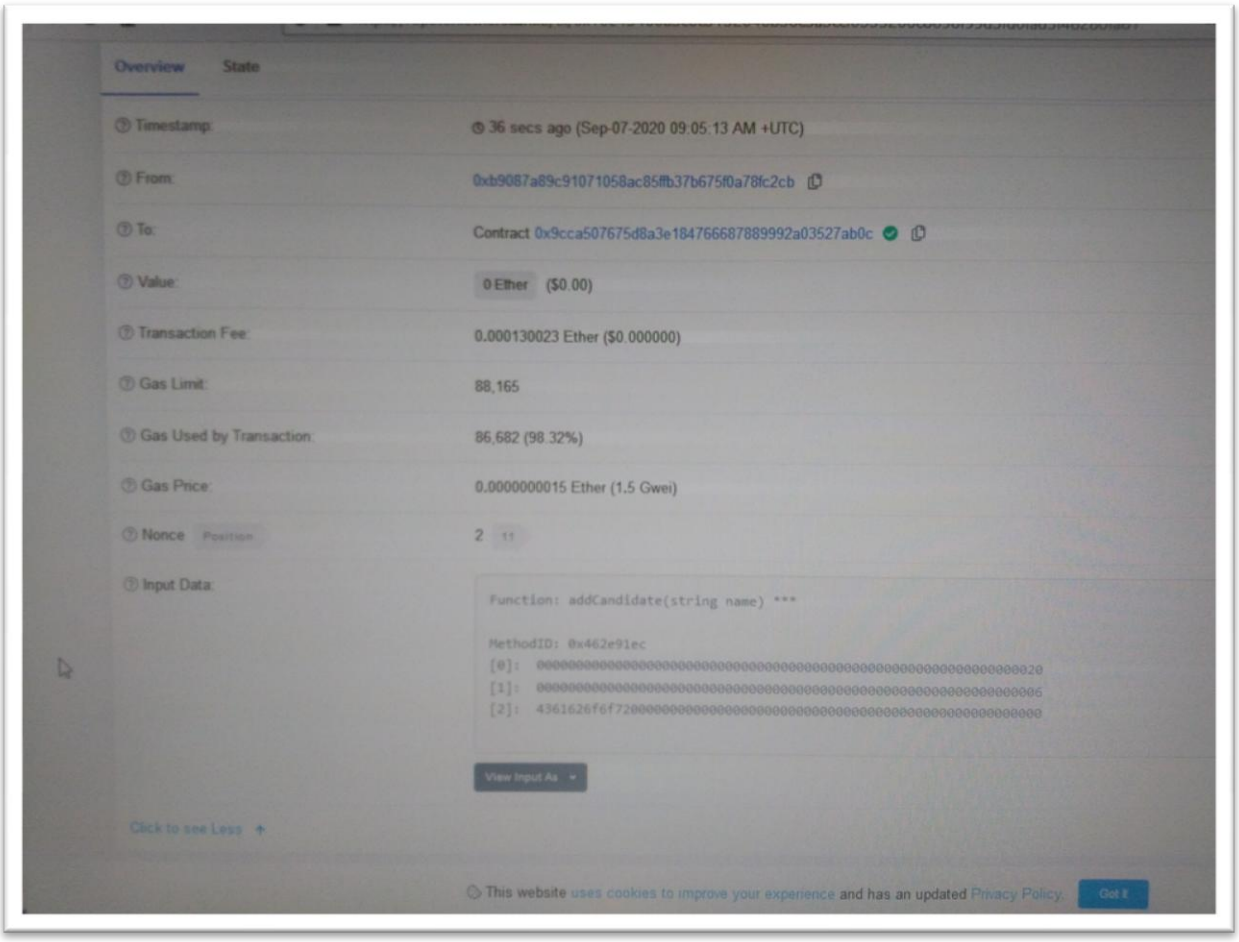


On cherche ensuite à ajouter un user au contrat nommé caboor :

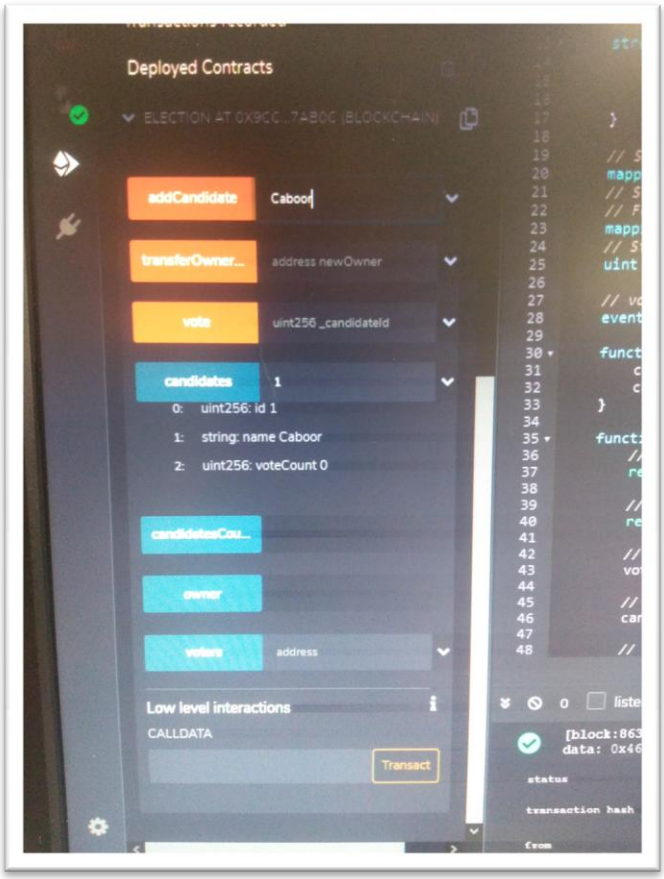


Le résultat de transaction donne :

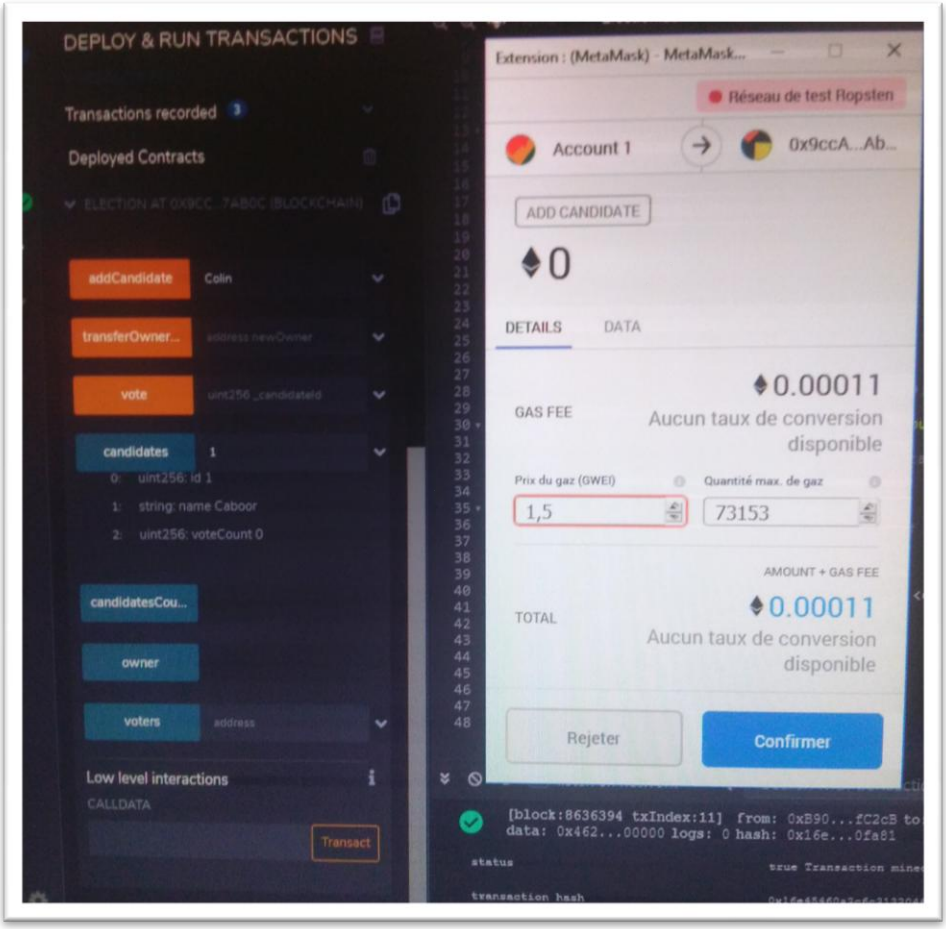




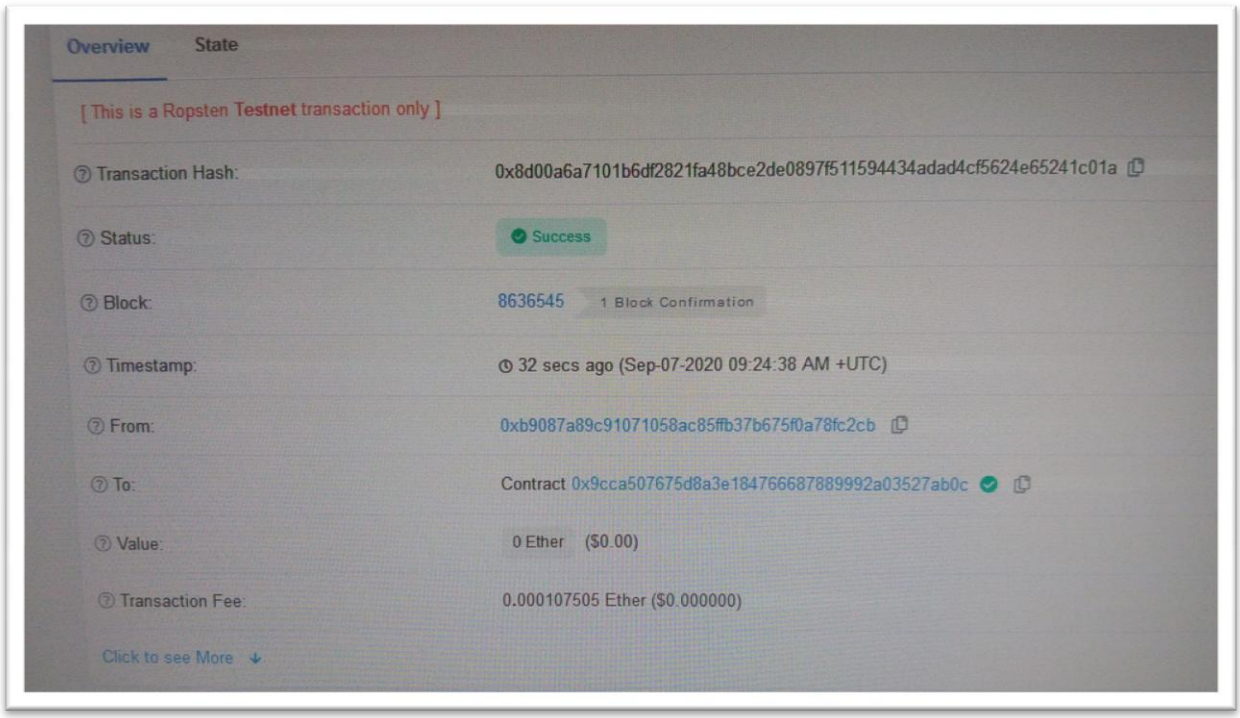
On peut alors vérifier sur remix la bonne création du user. Son id est 1. On voit notamment qu'il ne possède pour le moment aucun vote :

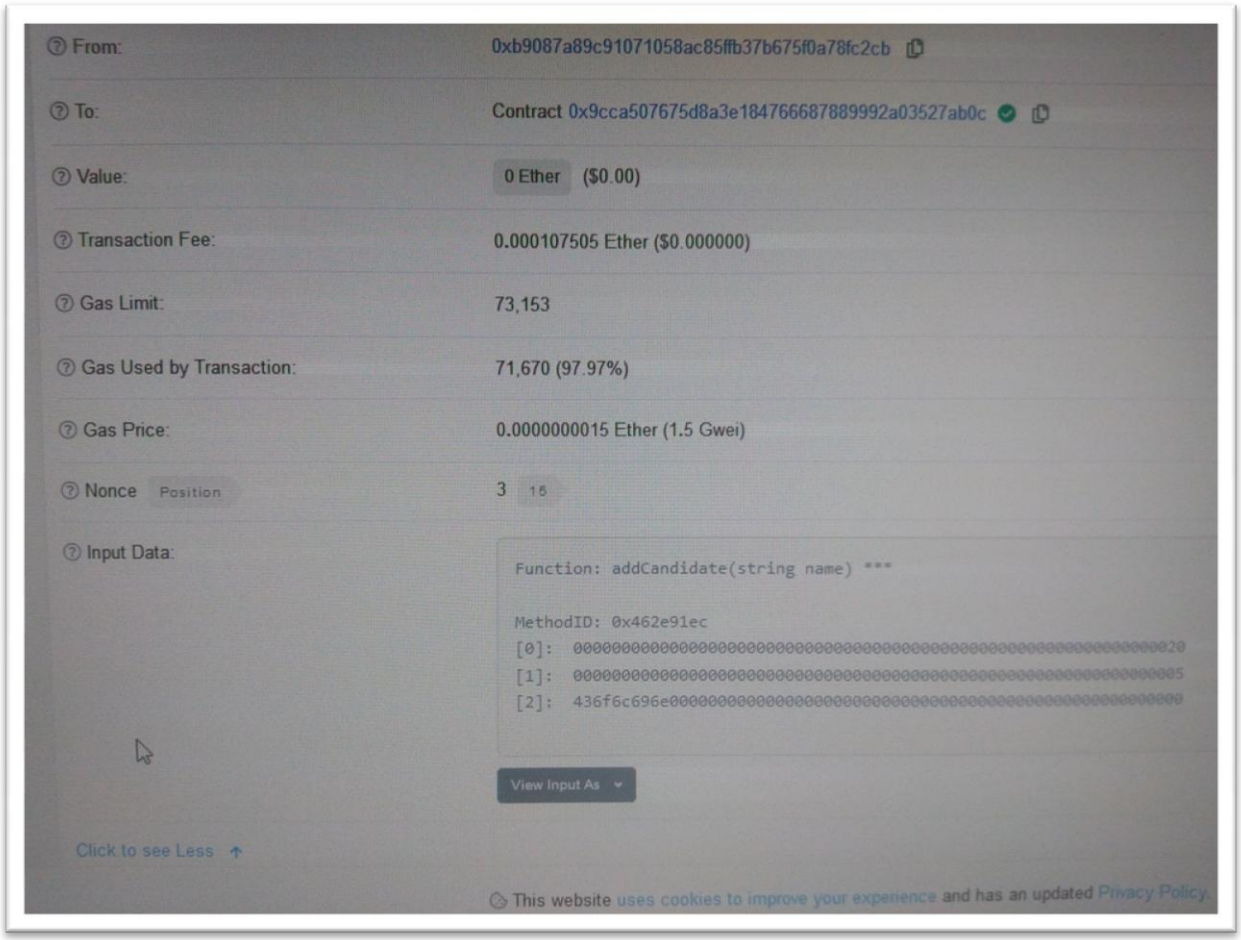


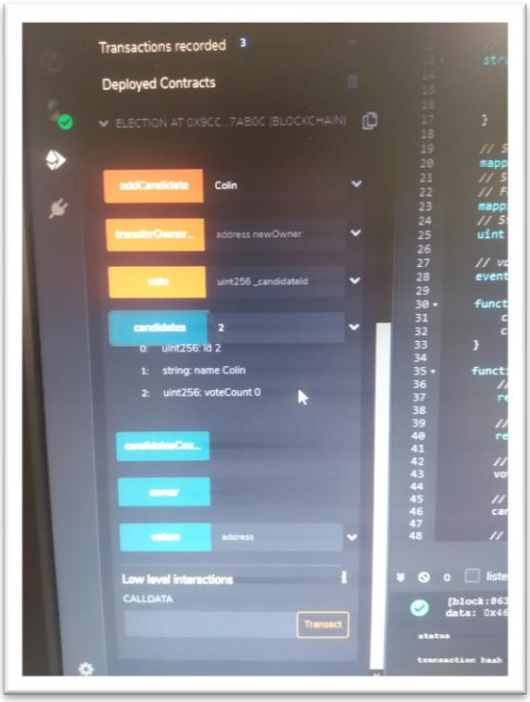
On ajoute ensuite un deuxième user nommé colin, qui aura alors un id 2:



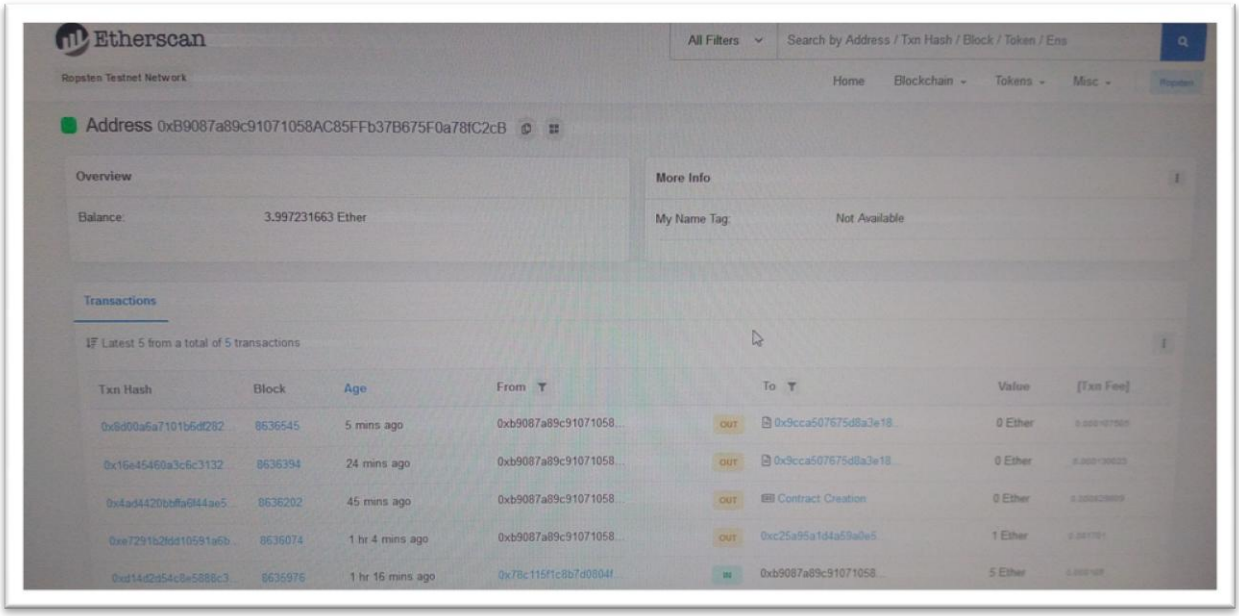
Son résultat de transaction donne :



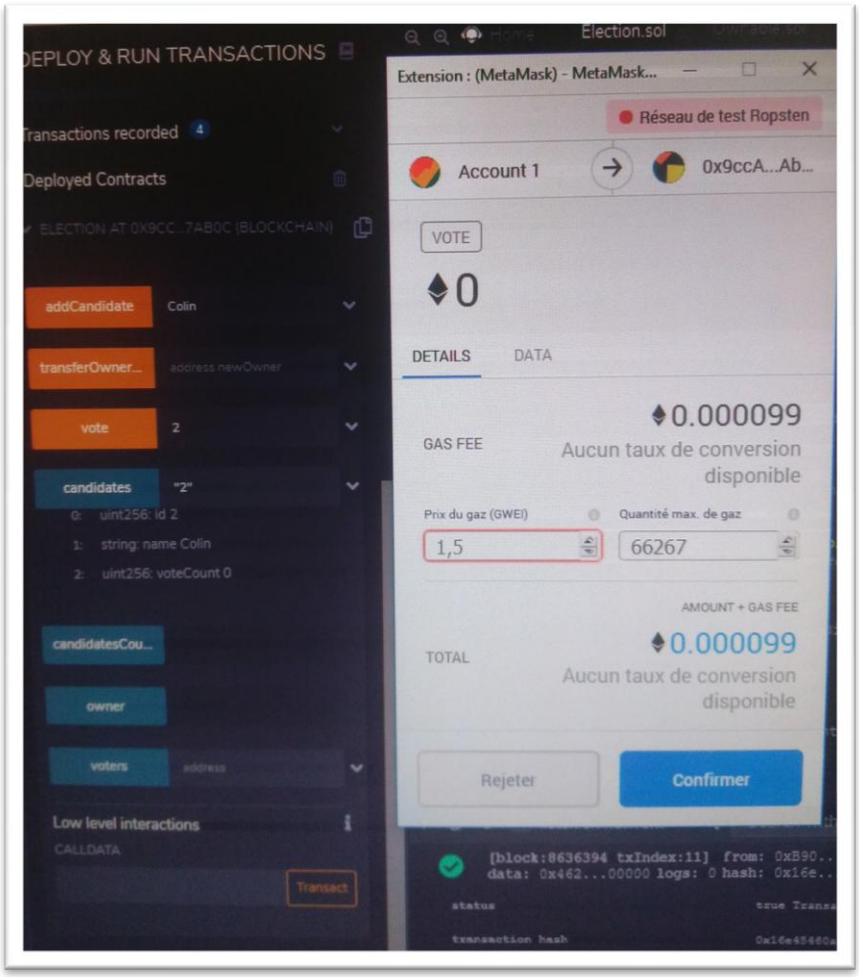




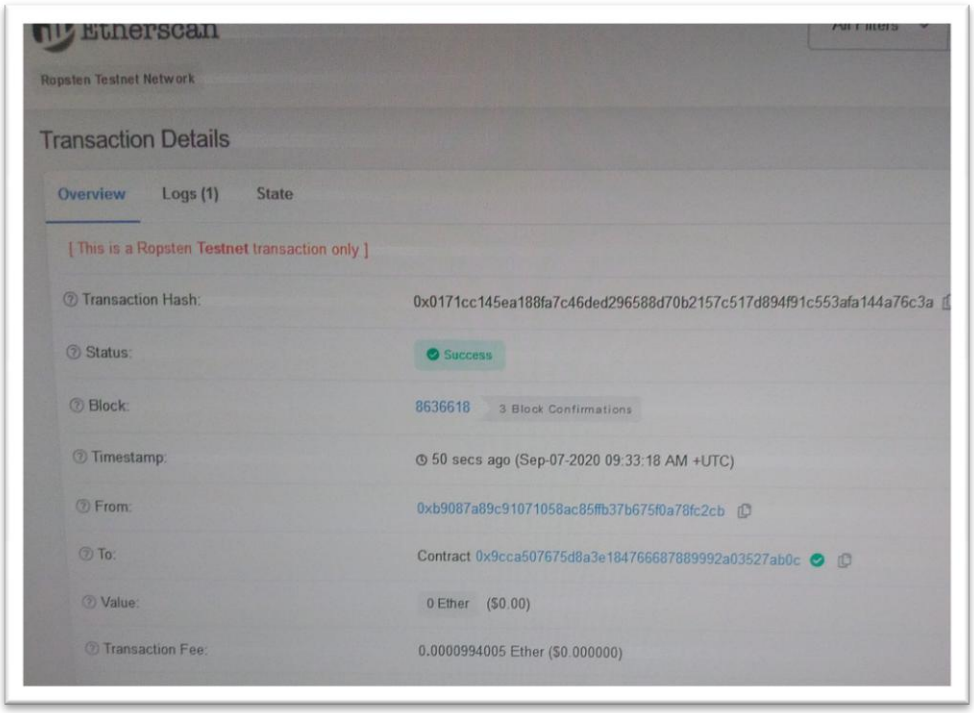
On peut alors récupérer l'adresse du propriétaire du contrat qui ici est 0xB9087a89c91071058AC85FFb37B675F0a78fC2cB :

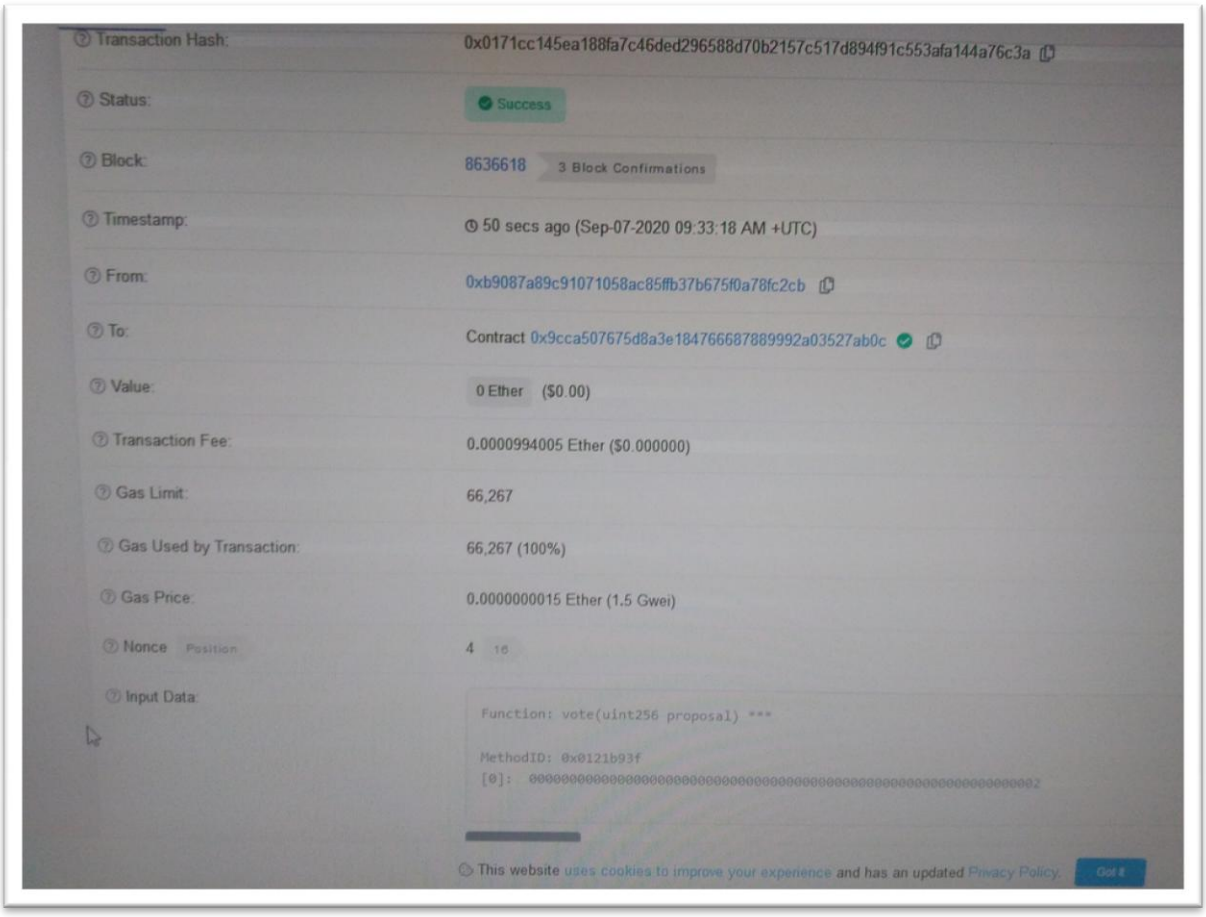


On peut aussi ajouter un vote à un utilisateur :

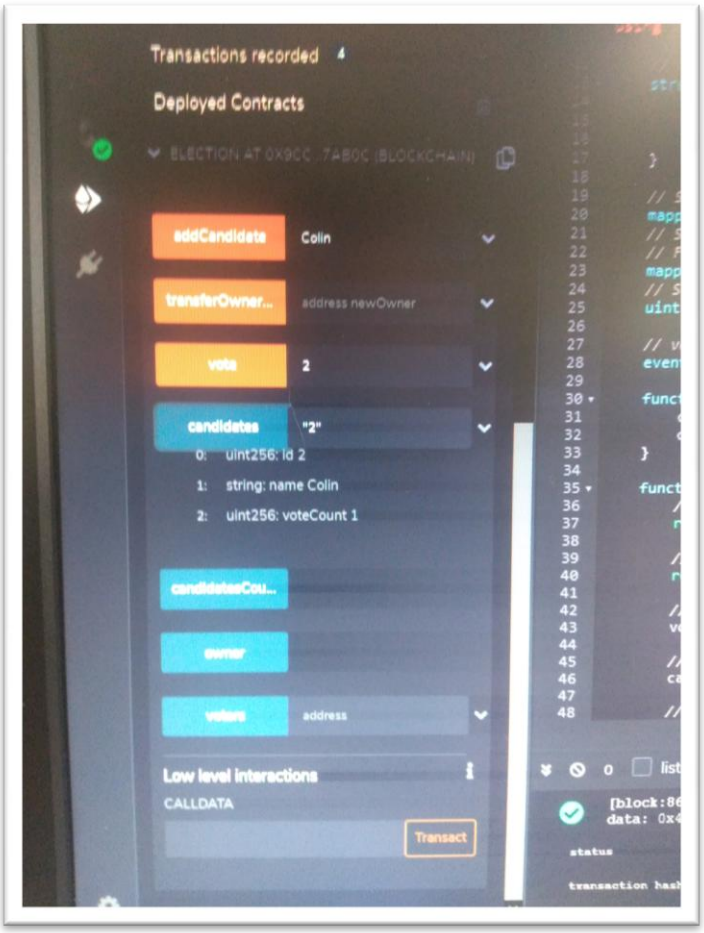


Résultat de transaction :

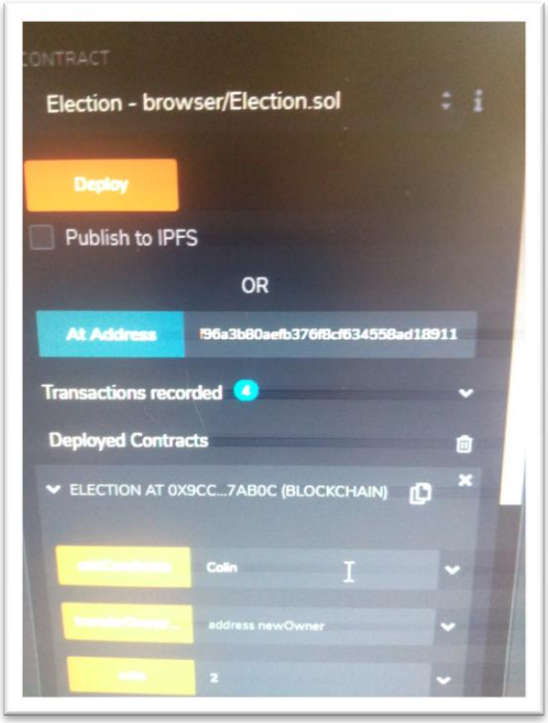


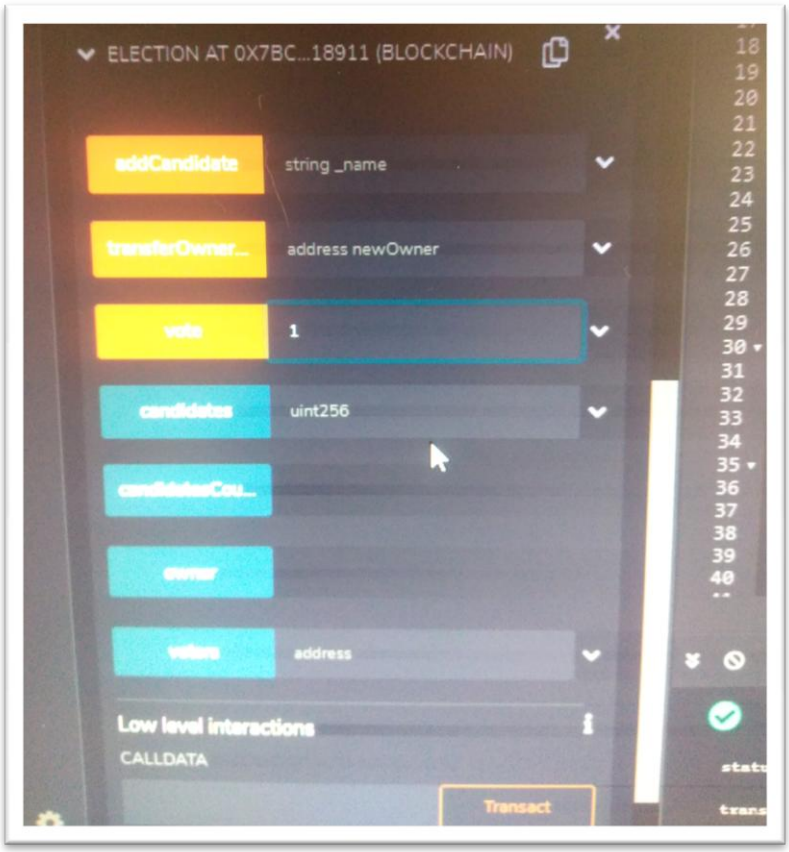


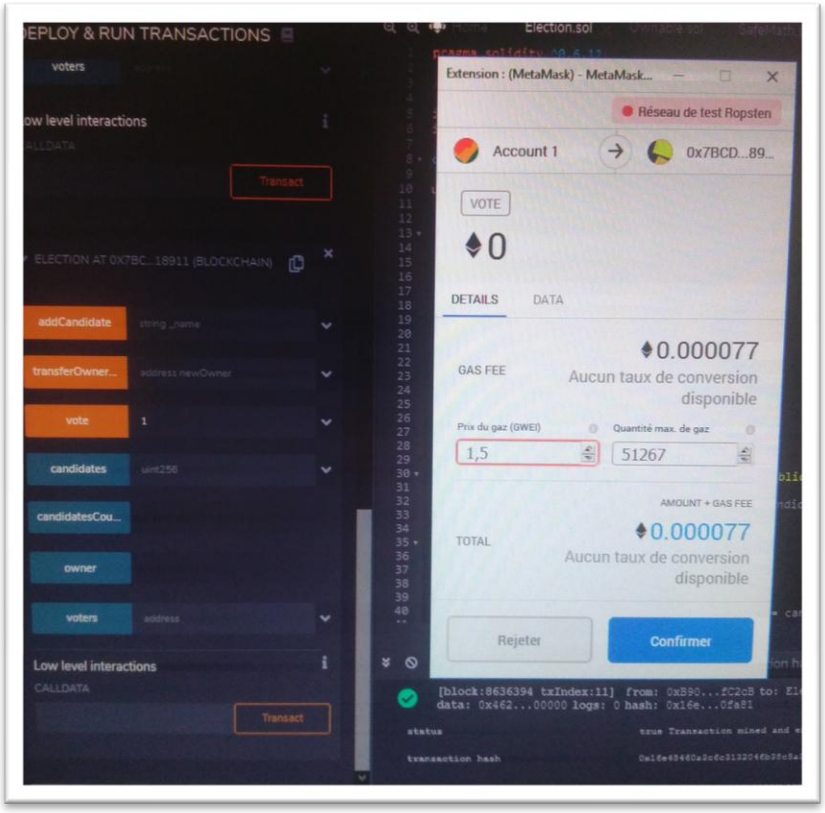
De même, pour ajouter un vote à l'utilisateur 2 :



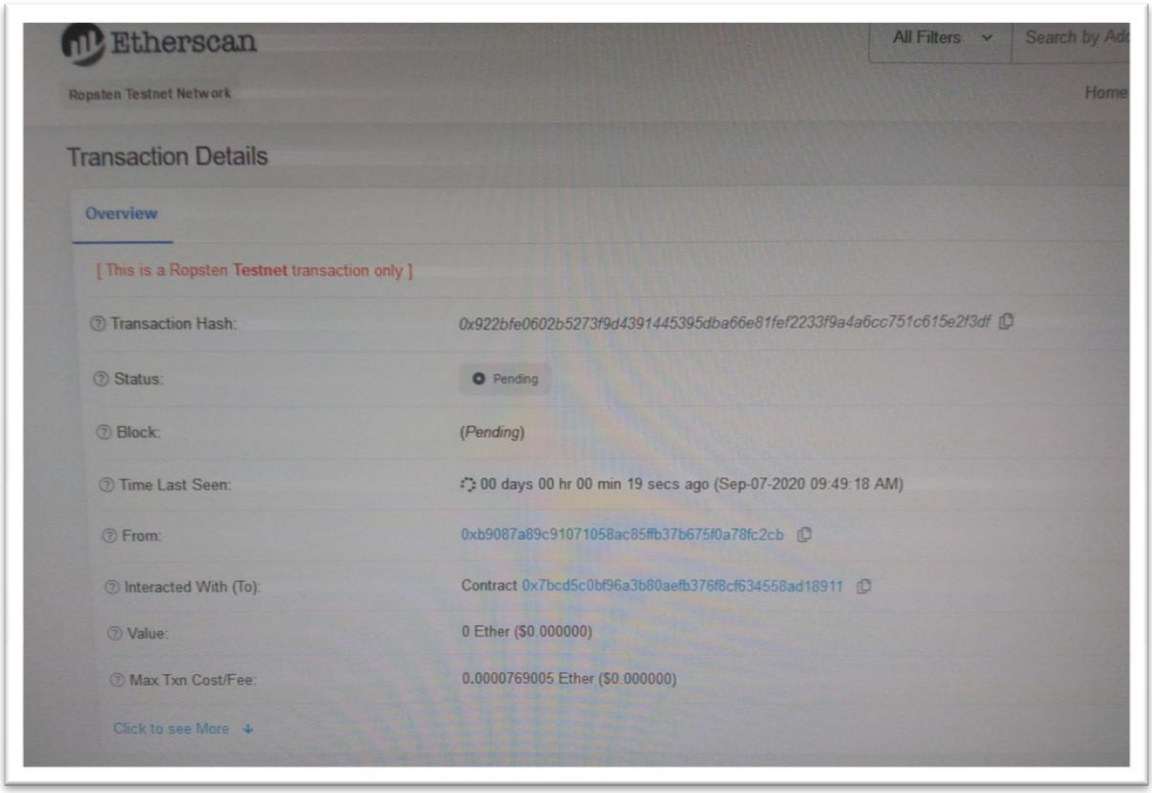
Le but ensuite et d'envoyer des votes pour le contrat d'un camarade. Pour cela, on renseigne sa clé et procède de la même manière :

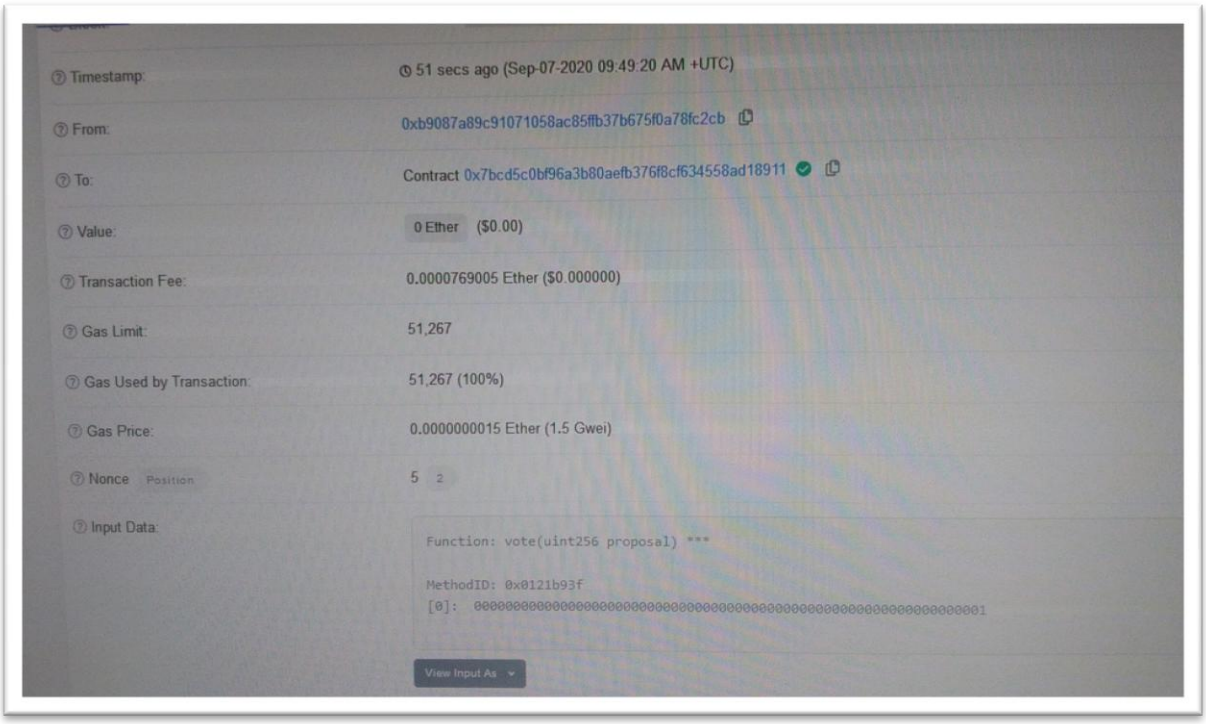




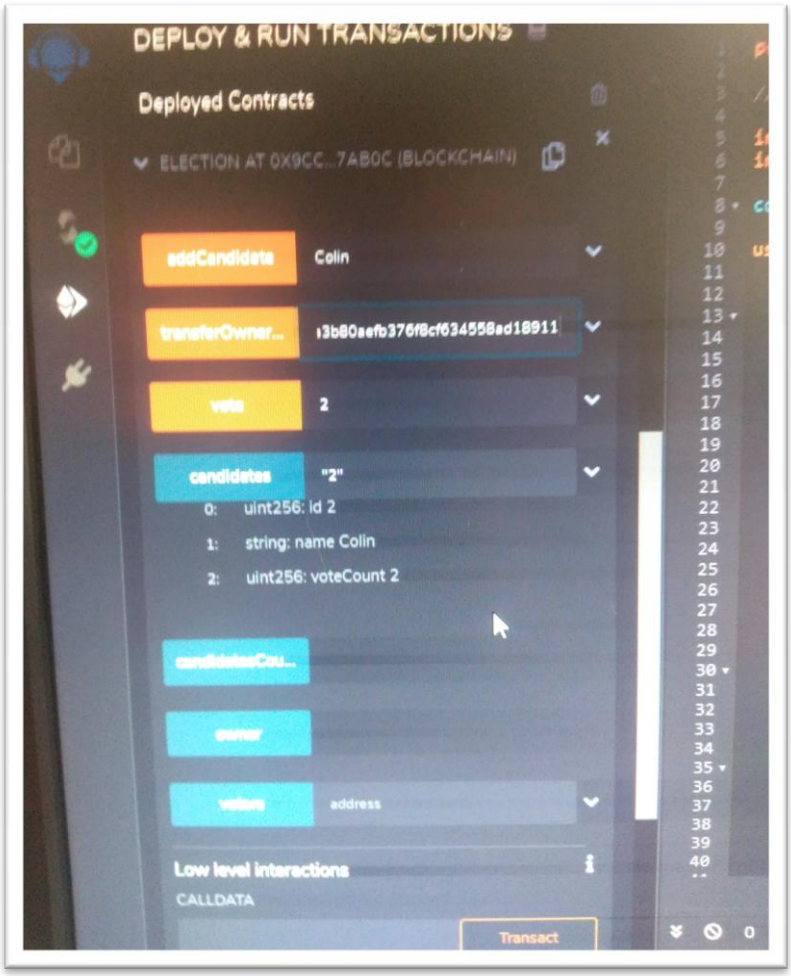


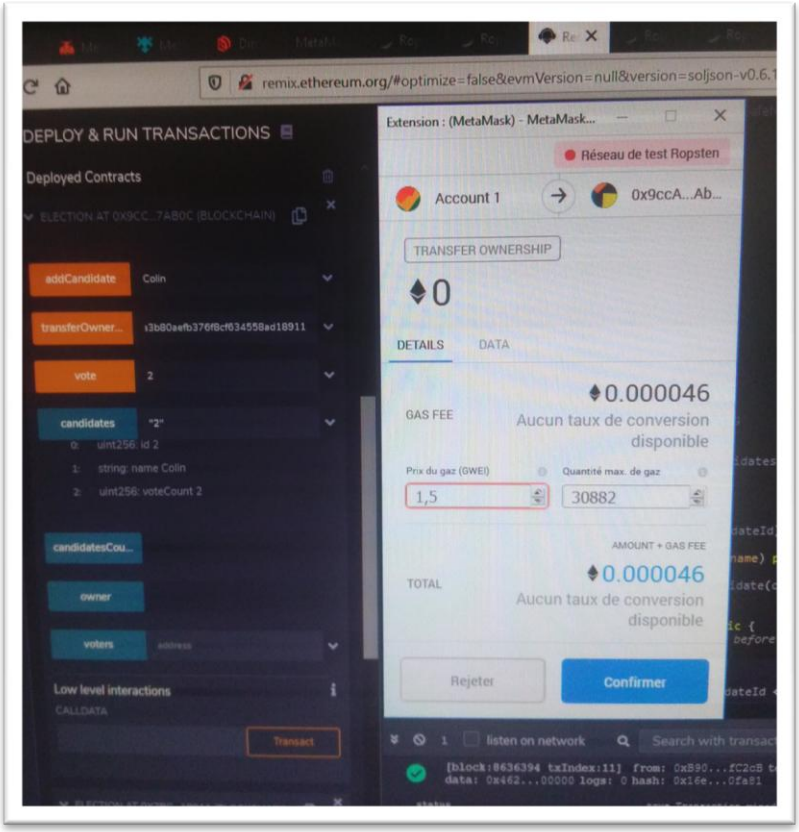
Résultat de transaction :

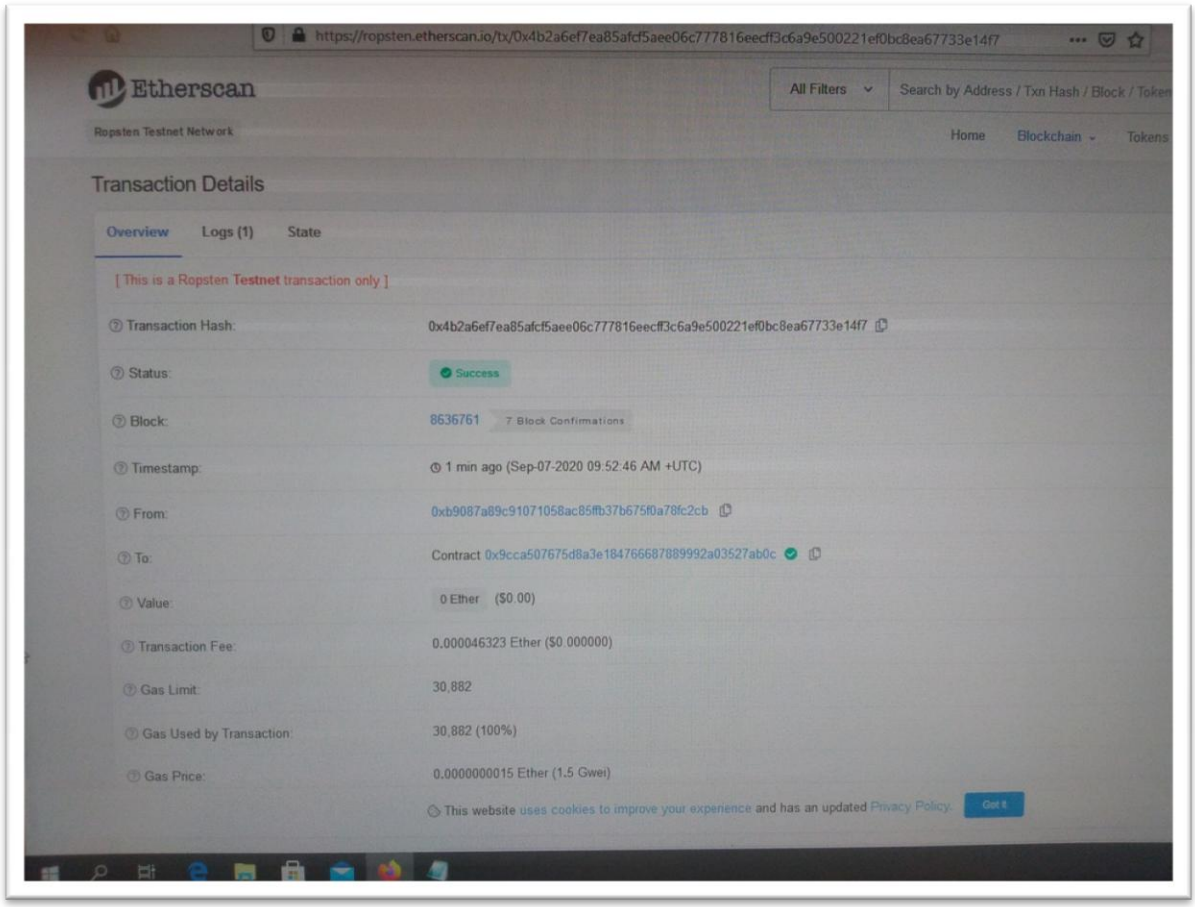


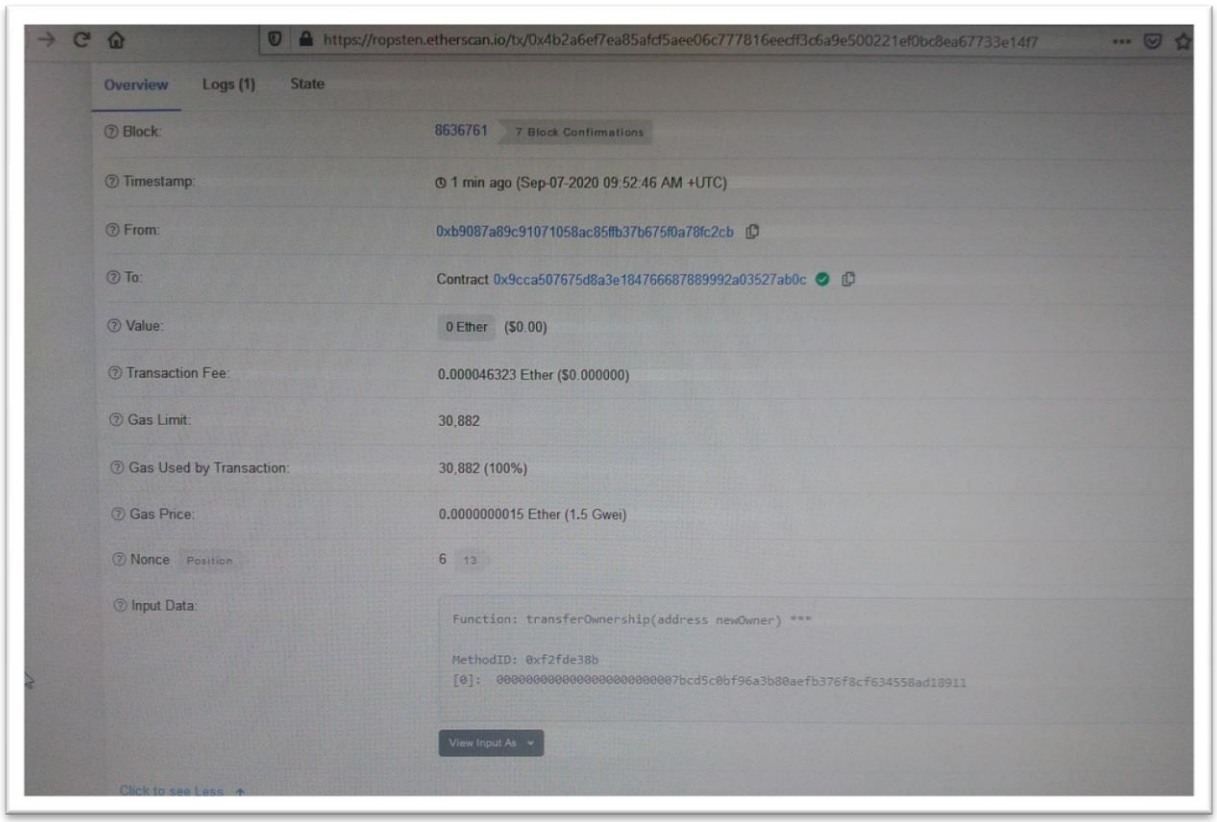


Idem pour le deuxième candidat :

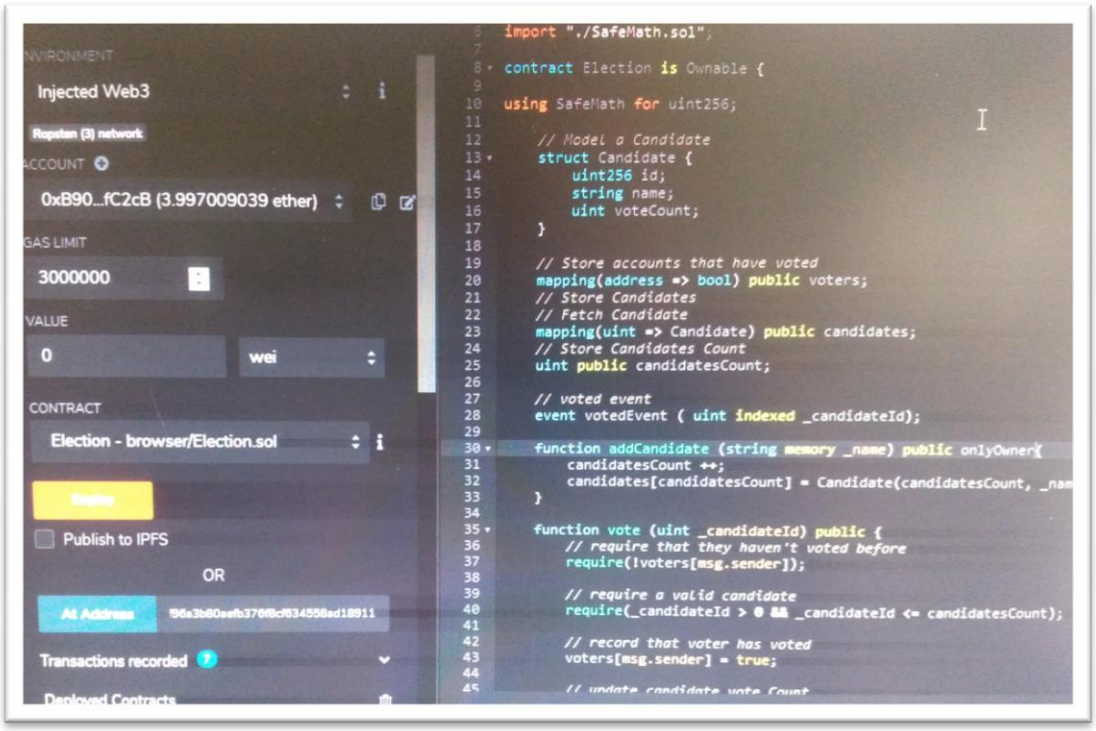

DEPLOY & RUN TRANSACTIONS
Deployed Contracts
ELECTION AT 0X9CC...7AB0C (BLOCKCHAIN)
addCandidate
Colin
transferOwner...
vote
2
candidates
"2"
0: uint256: id 2
1: string: name Colin
2: uint256: voteCount 2
owner
voters
address
Low level interactions
CALLDATA
Transact


remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.6.1
DEPLOY & RUN TRANSACTIONS
Deployed Contracts
Extension : (MetaMask) - MetaMask...
Réseau de test Ropsten
Account 1
0x9ccA...Ab...
TRANSFER OWNERSHIP
0
DETAILS
DATA
0.000046
GAS FEE
Aucun taux de conversion disponible
Prix du gaz (GWEI)
Quantité max. de gaz
1,5
30882
AMOUNT + GAS FEE
TOTAL
0.000046
Aucun taux de conversion disponible
Rejeter
Confirmer
listen on network
Search with transac





Enfin, si l'on veut pouvoir gérer les users en fonction du propriétaire du contrat :



Condition vérifiant que le owner du contrat est bien celui qui veut modifier les candidats.

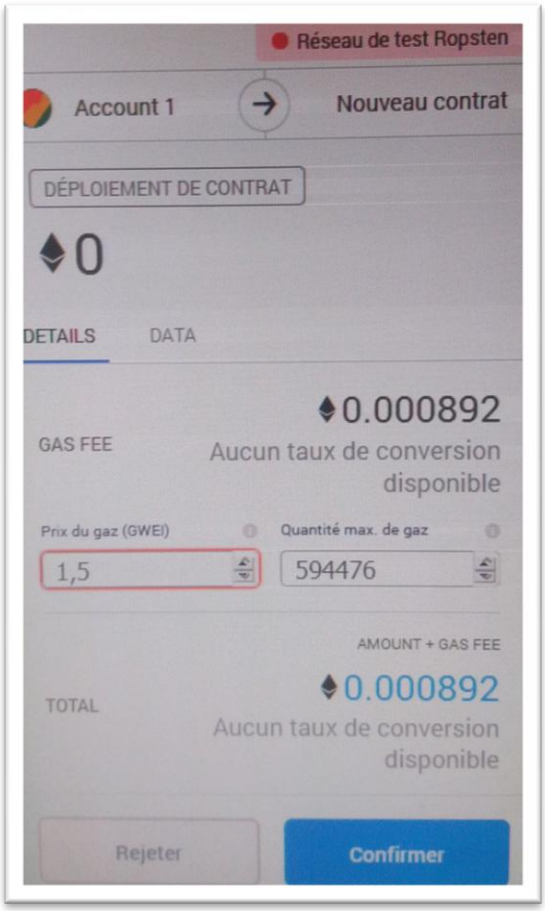
function addCandidate (string memory _name) public onlyOwner{

candidatesCount ++;

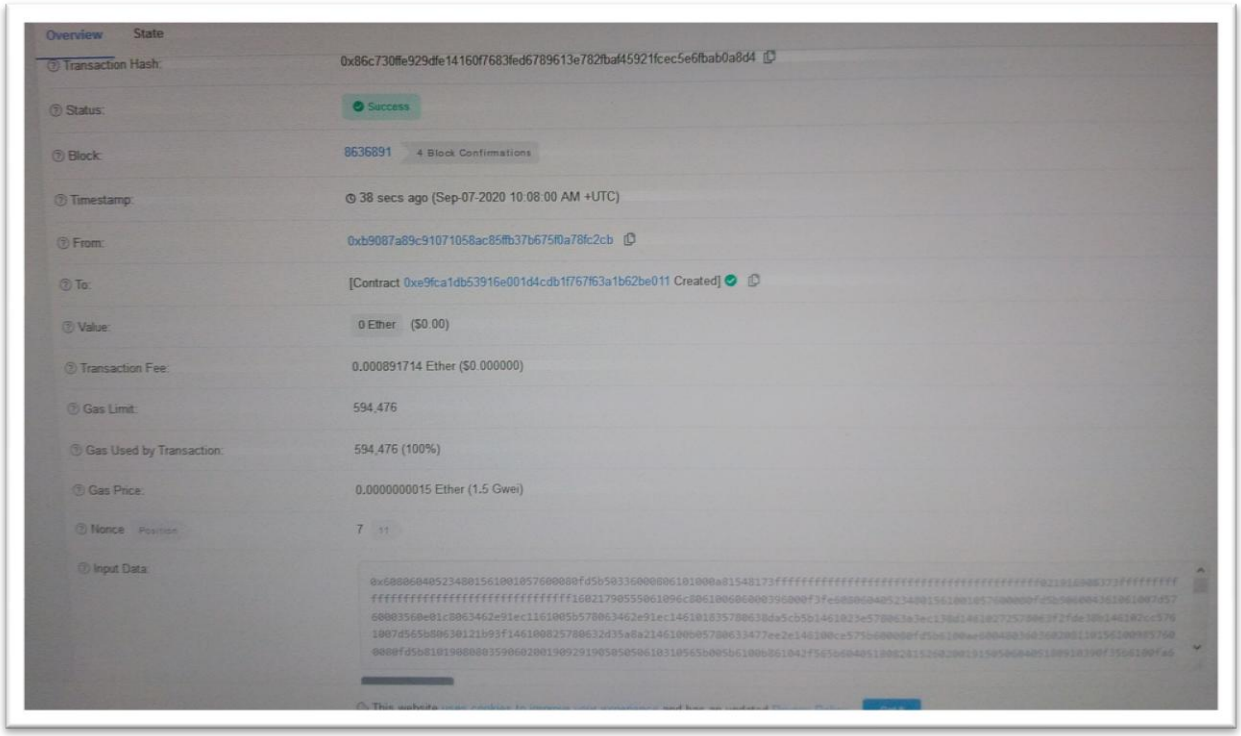
candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);

}

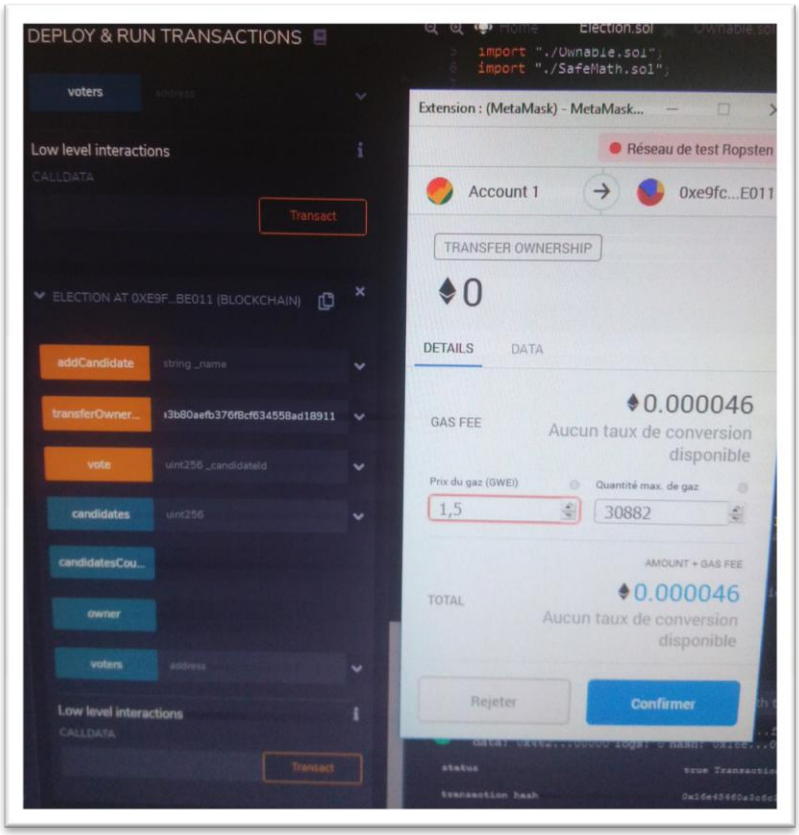
On n'a pas besoin de recompiler le code car on est en autocompile mais il faut cependant le re distribuer, il vient :

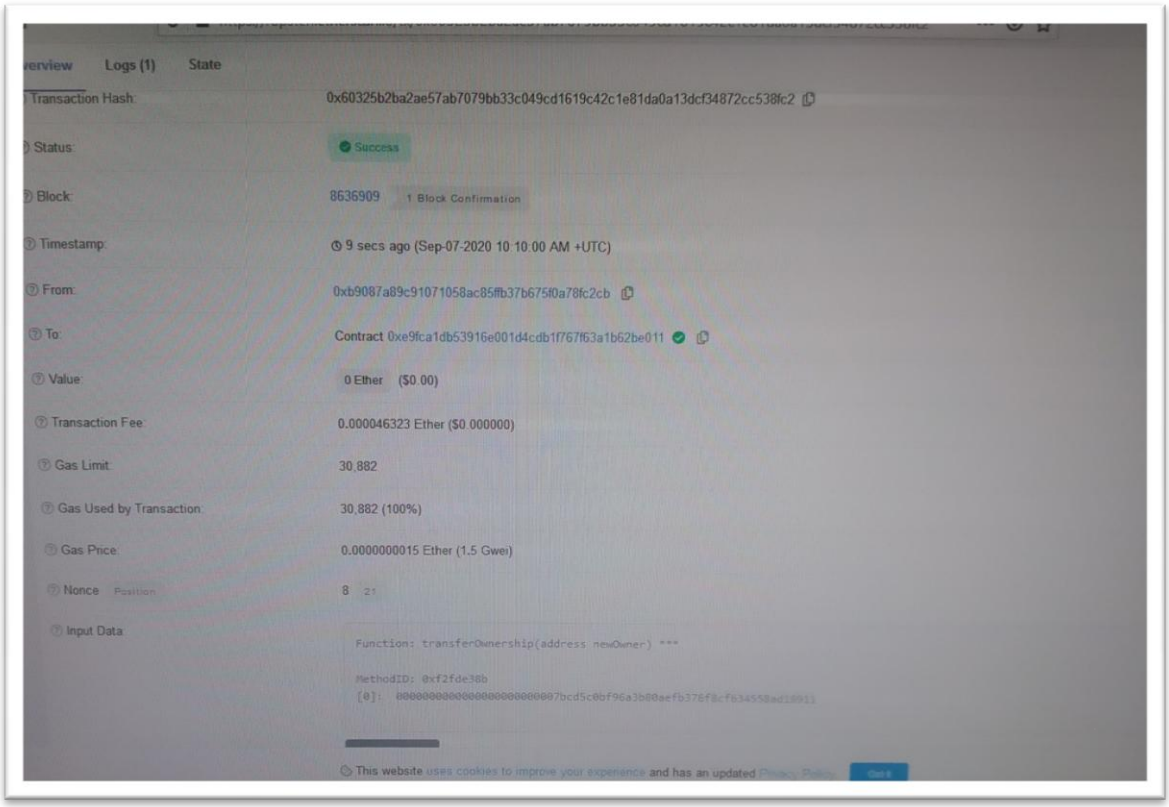


Résultat de distribution :

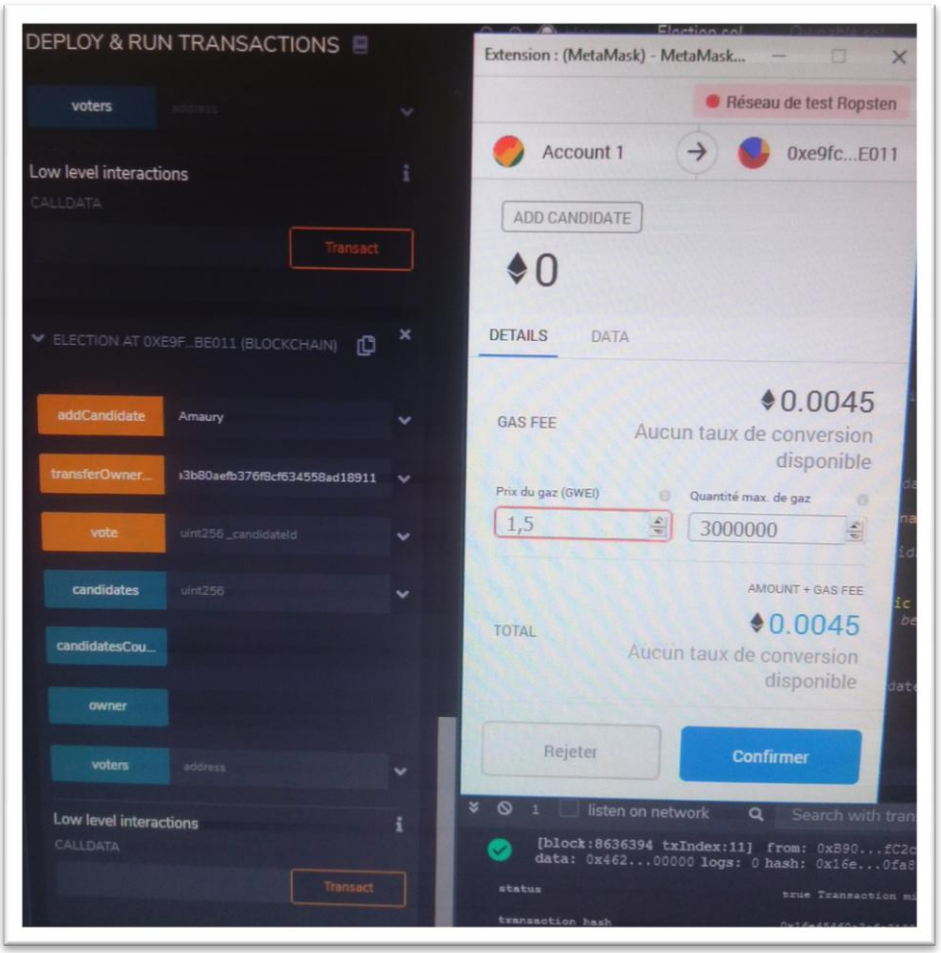


Enfin, nous avons vu comment changer le propriétaire d'un contrat :

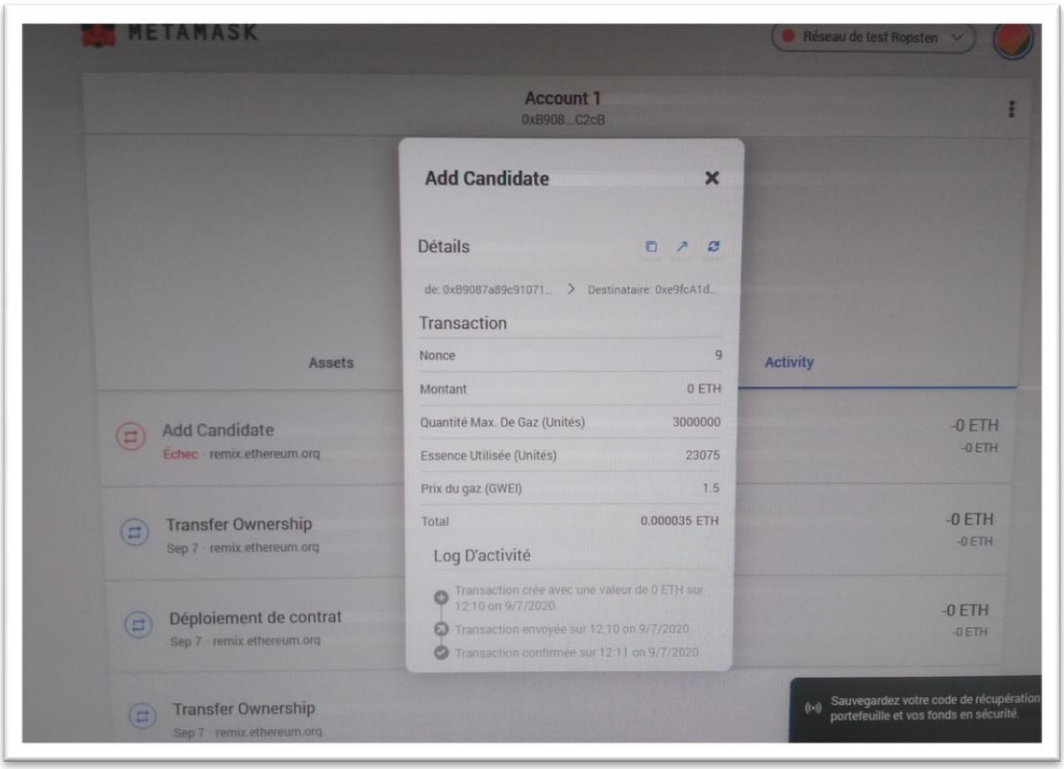


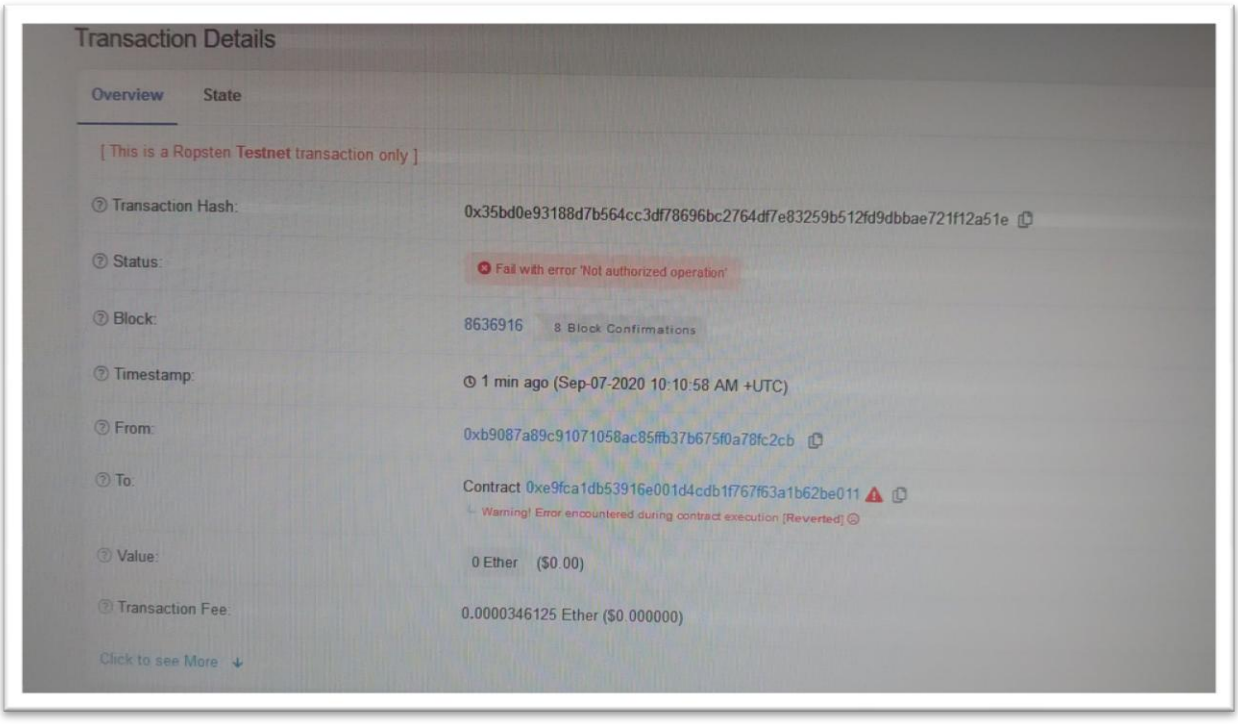


Une fois effectué, on fait le test pou savoir si un utilisateur autre que le propriétaire du contrat peut créer un nouvel utilisateur :



On voit alors que la transaction à échoué :


METAMASK
Réseau de test Ropsten
Account 1
0xB908...C2cB
Add Candidate
Détails
de: 0xB9087a89c91071... > Destinataire: 0xe9fcA1d...
Transaction
Nonce 9
Montant 0 ETH
Quantité Max. De Gaz (Unités) 3000000
Essence Utilisée (Unités) 23075
Prix du gaz (GWEI) 1.5
Total 0.000035 ETH
Log D'activité
Transaction crée avec une valeur de 0 ETH sur 12:10 on 9/7/2020.
Transaction envoyée sur 12:10 on 9/7/2020.
Transaction confirmée sur 12:11 on 9/7/2020.
Assets
Activity
Add Candidate
Échec · remix.ethereum.org
-0 ETH
-0 ETH
Transfer Ownership
Sep 7 · remix.ethereum.org
-0 ETH
-0 ETH
Déploiement de contrat
Sep 7 · remix.ethereum.org
-0 ETH
-0 ETH
Transfer Ownership
Sep 7 · remix.ethereum.org
Sauvegardez votre code de récupération portefeuille et vos fonds en sécurité.


Transaction Details
Overview
State
[This is a Ropsten Testnet transaction only]
Transaction Hash: 0x35bd0e93188d7b564cc3df78696bc2764df7e83259b512fd9dbbae721f12a51e
Status: Fail with error 'Not authorized operation'
Block: 8636916 8 Block Confirmations
Timestamp: 1 min ago (Sep-07-2020 10:10:58 AM +UTC)
From: 0xb9087a89c91071058ac85ffb37b675f0a78fc2cb
To: Contract 0xe9fca1db53916e001d4cdb1f767f63a1b62be011
Warning! Error encountered during contract execution [Reverted]
Value: 0 Ether ($0.00)
Transaction Fee: 0.0000346125 Ether ($0.000000)
Click to see More