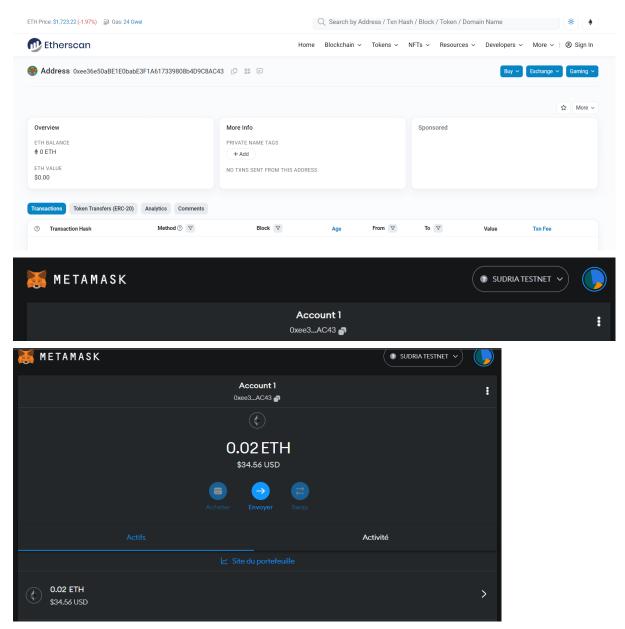
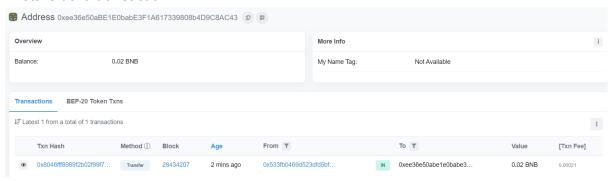
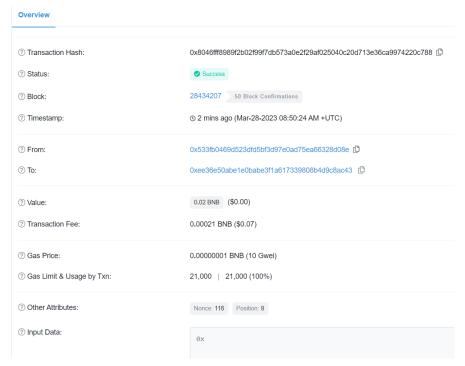
TRAVAUX PRATIQUE : Développer, Déployer et Interagir avec un contrat intelligent sur Ethereum

1. Prise en main des outils Remix et Metamask

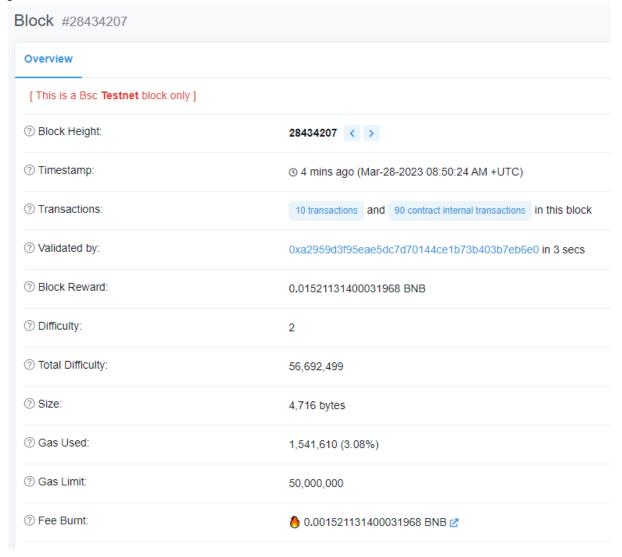


#### f.Détails de la transaction

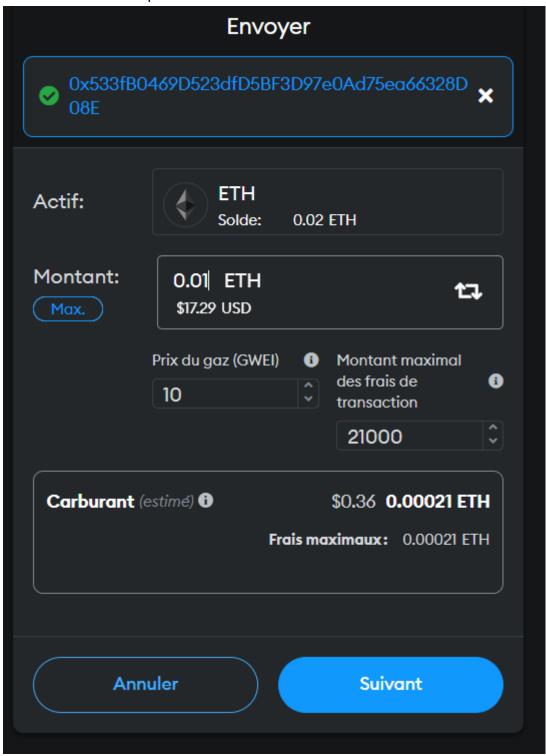


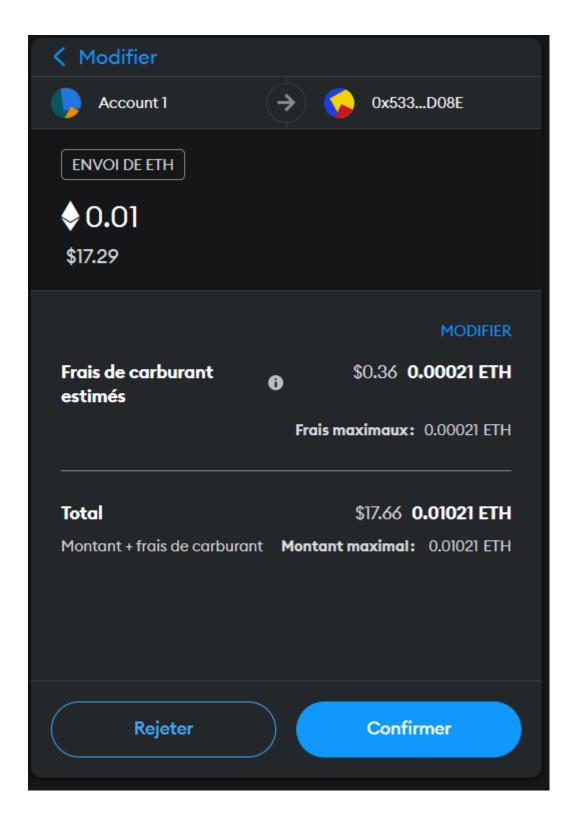


# g.Numéro de Block et transaction

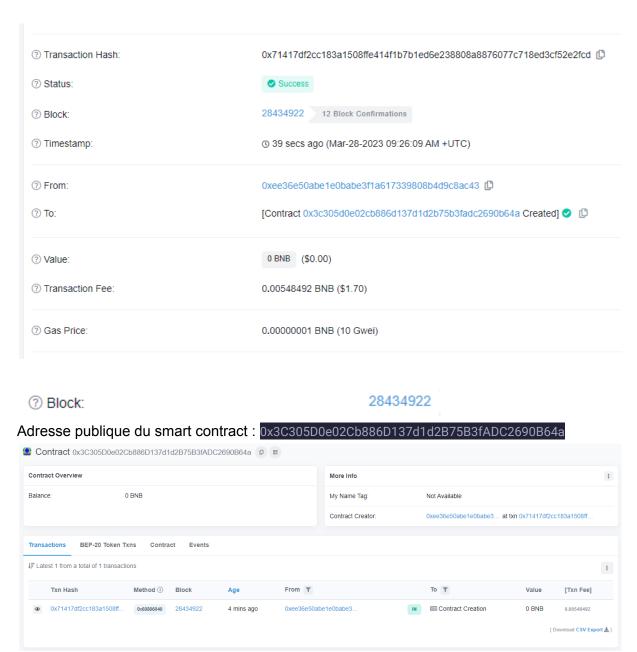


h.Génération de notre première transaction Ethereum



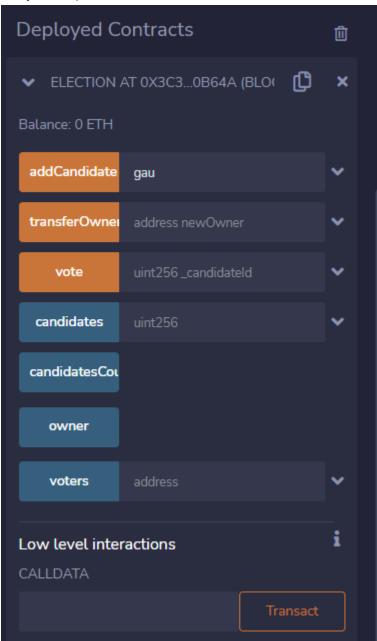


- I. ABI et Byte code du contrat: cf voir les fichiers
- M.Déploiement du smart contract "Election.sol"

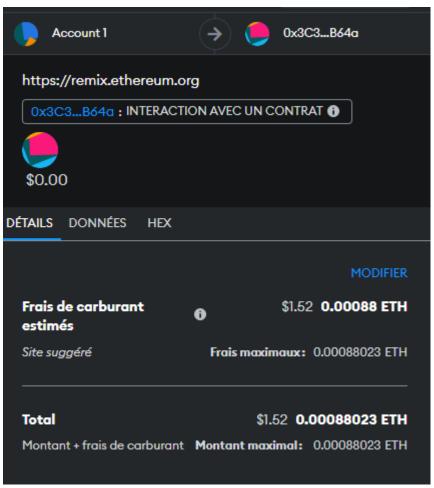


Justification des frais: La puissance de calcul demandée n'est pas la même, donc les frais de transaction sont différents.

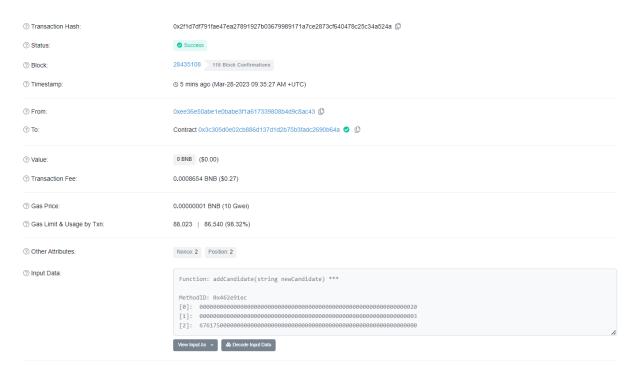
# N. Ajout du premier candidat



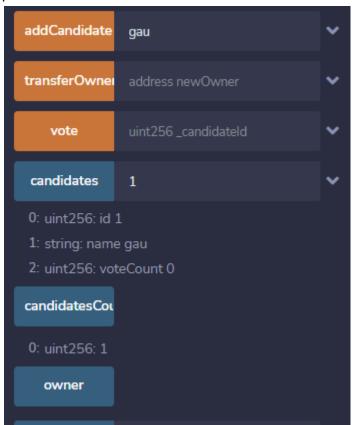
#### O.Génération de la transaction



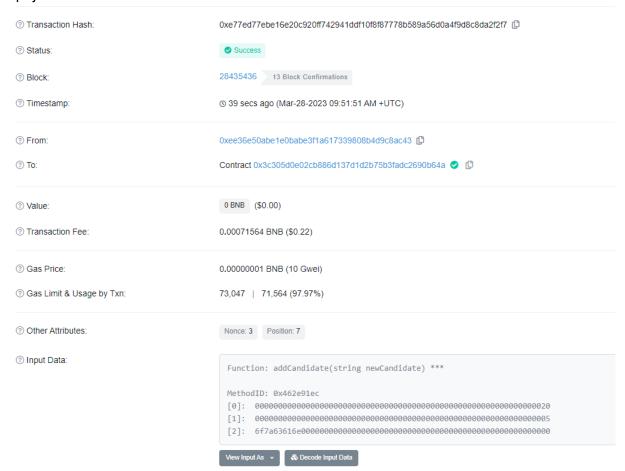




# p.Consultation du "CandidateID"



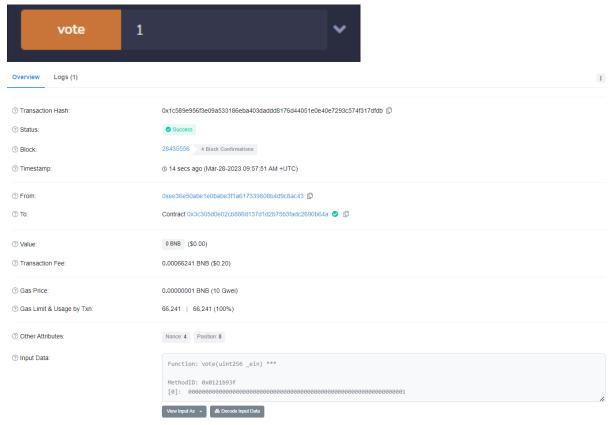
# q.Ajout du second candidat



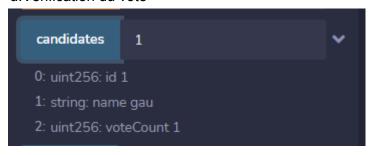
#### r.Consultation de la valeur du second CandidateID



s.L'adresse du propriétaire du contract: 0xee36e50abe1e0babe3f1a617339808b4d9c8ac43 t.Réalisation du premier vote

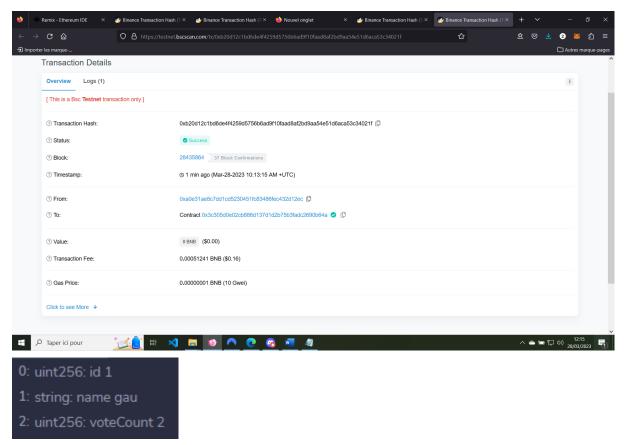


### u. Vérification du vote



٧.

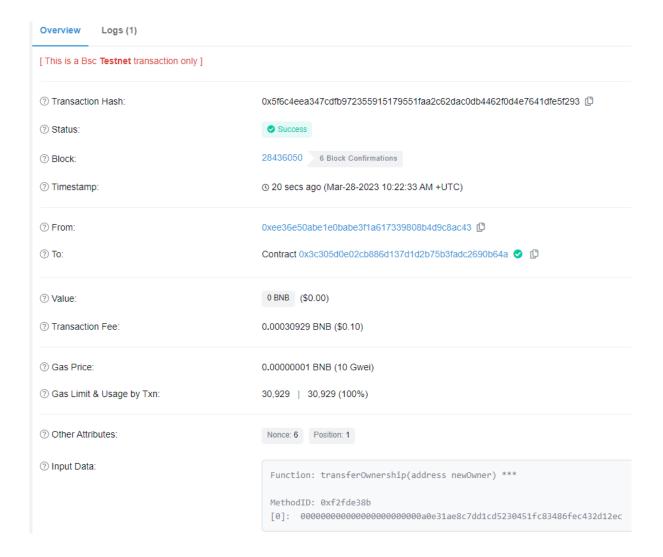
Mon contrat: 0x3C305D0e02Cb886D137d1d2B75B3fADC2690B64a

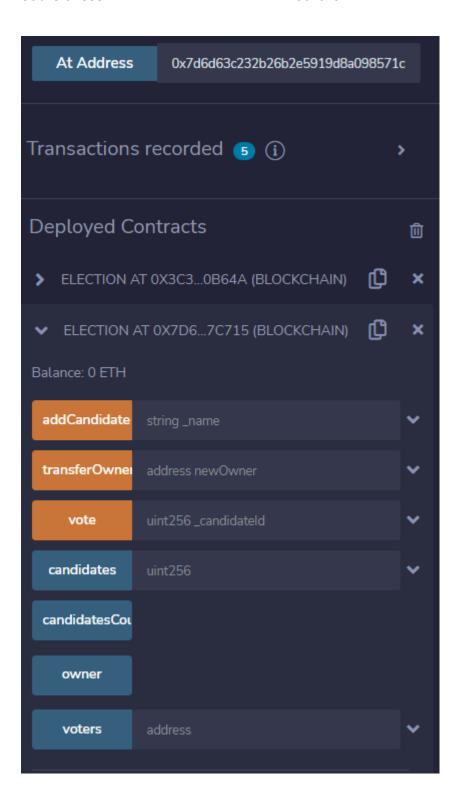


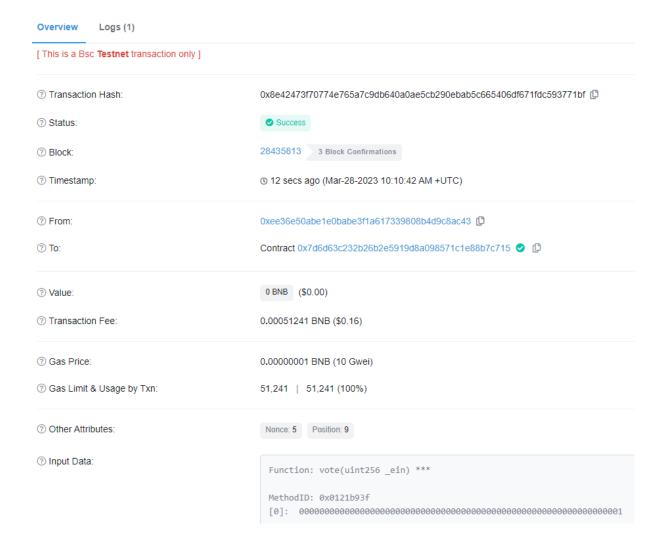
w.Réalisation du transfert

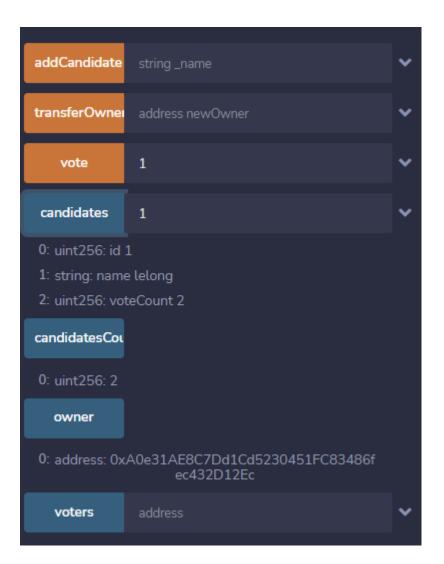
Contrat Antoine Lelong: 0x7D6D63C232b26b2E5919D8A098571C1e88b7C715











x. On peut faire en sorte que l'utilisation soit uniquement réservée au owner du contrat. y.

```
/**
  * @dev Throws if called by any account other than the owner.
  */
modifier onlyOwner() {
    require(msg.sender == owner, "Not authorized operation");
    _;
}
```

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