

TP BLOCKCHAIN

Cécilia CARIUS A21

a)b)c)d)e)

faucet.dimensions.network

Ropsten Testnet ETH Faucet

Enter Your Ropsten Address

0x542b817700C2A772E0DE9666731F6B5D39734677

Send Ropsten ETH

397129 ETH left in Faucet. Gas Limit 400k

Congratulations, your address has been added to the queue.

Pending

Currency	Address	Amount	Date/Time
ETH-Ropsten	0xC8134f8C874359e9a8EC0005aF6a409446a59A	5	Sept. 7, 2020, 8:11 a.m.
ETH-Ropsten	0x50AA2C1CAa90853f9605b38f377404a7D302d	5	Sept. 7, 2020, 8:11 a.m.
ETH-Ropsten	0xc0B607E491B0e5A82B0c33F48e6543D3c9968612	5	Sept. 7, 2020, 8:11 a.m.

METAMASK

Réseau de test Ropsten

Account 1 / ETH

5.9983 ETH

BUY ENVOYER

Envoyer des ETH
Sep 7 · To: 0x4415...d3ba -2 ETH

Receive
Sep 7 · From: 0x78c1...ee78 5 ETH

Receive
Sep 7 · From: 0x81b7...7647 1 ETH

CeciliaCrs/TP_smartcoi x TP 2 - Smart Contract x Remix - Ethereum IDE x Dimensions Cryptocur x MetaMask x Test Ether Faucet x Ropsten Transaction H x

ropsten.etherscan.io/bx/0x9eb40a9a9eeba85e6469ed1231baca06f17358a8d8122d350b2363e13f3f935

YouTube Traduire ES ESME-Sudria - Bien... Installez les outils d... Mon Drive - Googl... (1) WhatsApp Trader : 8 conseils p... UNIDAYS - Des offr... CACEIS. WebEx Ent... Conferencing Dial-L... Autres favoris

Etherscan

Ropsten Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Ropsten

Transaction Details

Overview State

[This is a Ropsten Testnet transaction only]

Transaction Hash:	0x9eb40a9a9eeba85e6469ed1231baca06f17358a8d8122d350b2363e13f3f935
Status:	Success
Block:	8636036 717 Block Confirmations
Timestamp:	1 hr 32 mins ago (Sep-07-2020 08:19:08 AM +UTC)
From:	0x78c115f1c8b7d0804bd3cf7995b030c512ee78
To:	0x542b817700c2a772e0de966673f8b5d39734677
Value:	5 Ether (\$0.00)
Transaction Fee:	0.000084 Ether (\$0.000000)

[Click to see More](#)

This website uses cookies to improve your experience and has an updated Privacy Policy. [Got it](#)

f)

Numéro du block : 8636036

CeciliaCrs/TP_smartcoi x TP 2 - Smart Contract x Remix - Ethereum IDE x Dimensions Cryptocur x MetaMask x Test Ether Faucet x Ropsten Transaction H x

ropsten.etherscan.io/bx/0x9eb40a9a9eeba85e6469ed1231baca06f17358a8d8122d350b2363e13f3f935

YouTube Traduire ES ESME-Sudria - Bien... Installez les outils d... Mon Drive - Googl... (1) WhatsApp Trader : 8 conseils p... UNIDAYS - Des offr... CACEIS. WebEx Ent... Conferencing Dial-L... Autres favoris

Etherscan

Ropsten Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Ropsten

Transaction Details

Overview State

[This is a Ropsten Testnet transaction only]

Transaction Hash:	0x9eb40a9a9eeba85e6469ed1231baca06f17358a8d8122d350b2363e13f3f935
Status:	Success
Block:	8636036 736 Block Confirmations
Timestamp:	1 hr 35 mins ago (Sep-07-2020 08:19:08 AM +UTC)
From:	0x78c115f1c8b7d0804bd3cf7995b030c512ee78
To:	0x542b817700c2a772e0de966673f8b5d39734677
Value:	5 Ether (\$0.00)
Transaction Fee:	0.000084 Ether (\$0.000000)

[Click to see More](#)

This website uses cookies to improve your experience and has an updated Privacy Policy. [Got it](#)

[This is a Ropsten Testnet block only]

Block Height:	8636036
Timestamp:	1 hr 37 mins ago (Sep-07-2020 08:19:08 AM +UTC)
Transactions:	8 transactions and 0 contract internal transaction in this block
Mined by:	0xd34912efb0e7f6daedb9390990d7ef623e01f4fa in 3 secs
Block Reward:	2.0053919195 Ether (2 + 0.0053919195)
Uncles Reward:	0
Difficulty:	553,086,213
Total Difficulty:	31,436,516,451,288,210
Size:	1,590 bytes
Gas Used:	214,004 (2.70%)
Gas Limit:	7,924,805
Extra Data:	poolin.com (Hex: 0x706f8f6c596e2e636f5d)

This website uses cookies to improve your experience and has an updated Privacy Policy. [Get it](#)

TP BLOCKCHAIN - Word (Produit sans licence)

g)

METAMASK Réseau de test Ropsten

Account 1 / ETH

5.9983 ETH

[BUY](#) [ENVOYER](#)

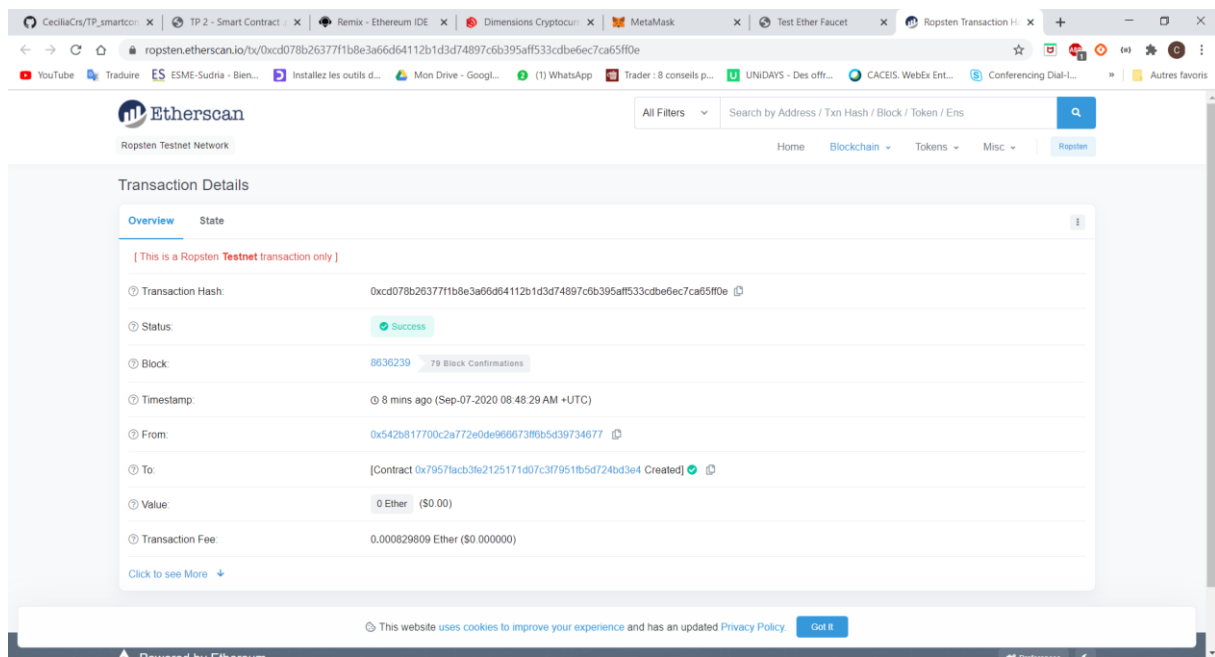
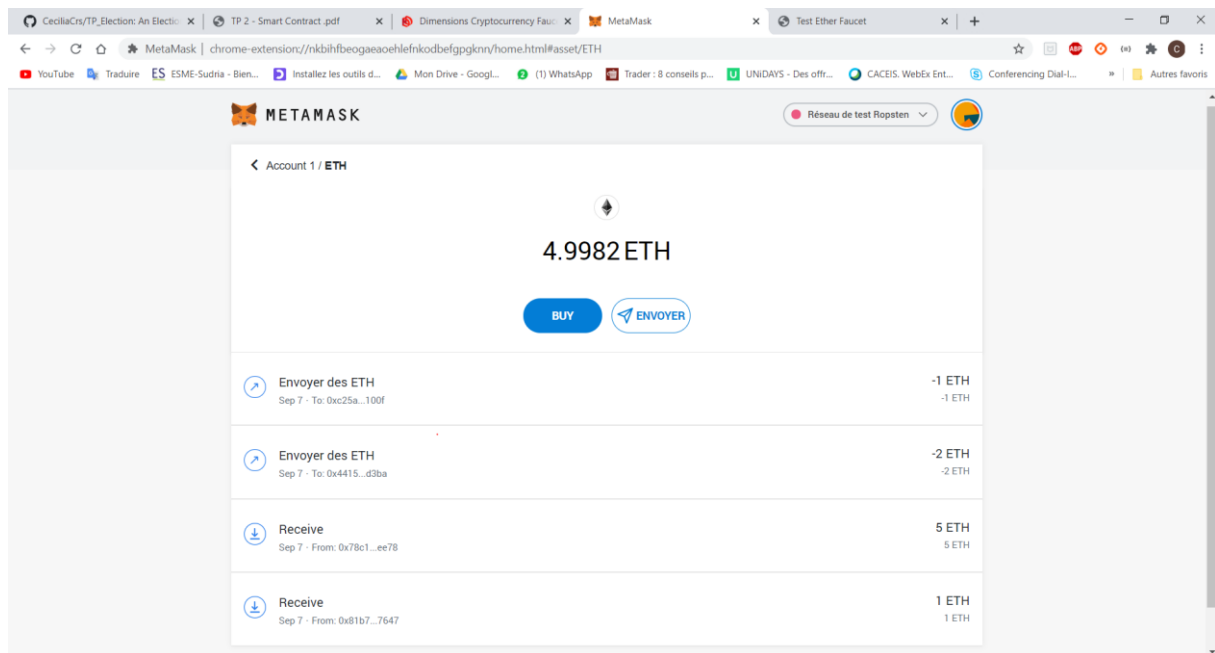
File d'attente (1)

Envoyer des ETH
En attente - To: 0xc25a...100f
[accélérer](#) [Annuler](#) -1 ETH -1 ETH

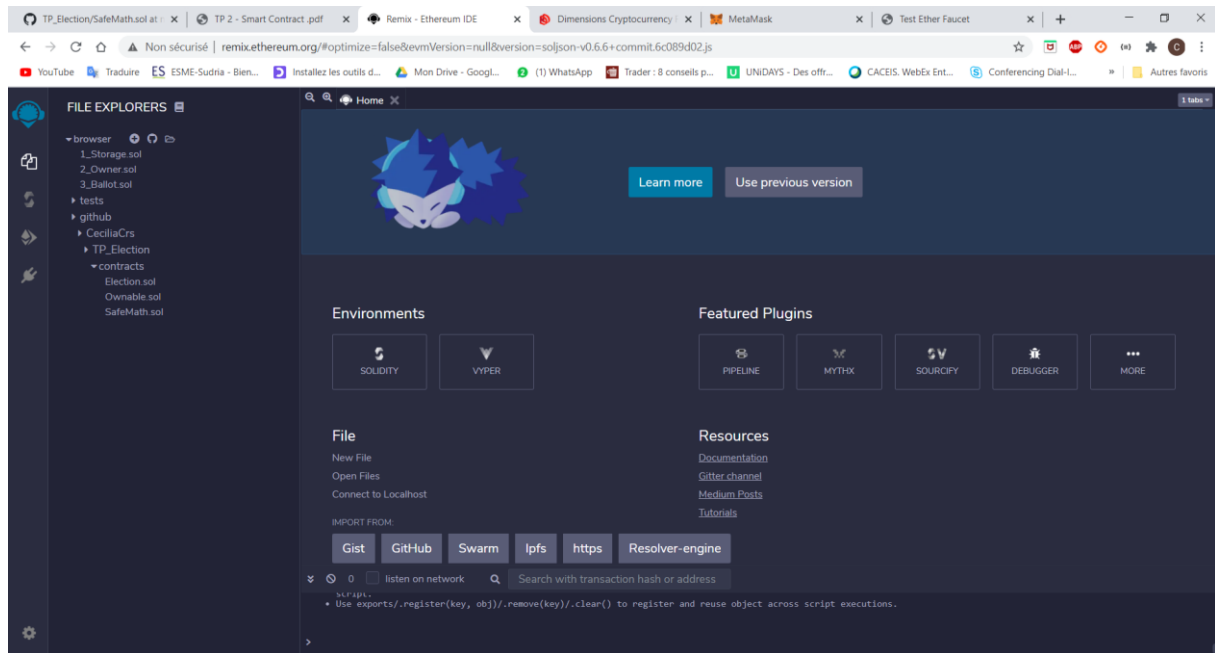
Historique

Envoyer des ETH
Sep 7 - To: 0x4415...d3ba -2 ETH -2 ETH

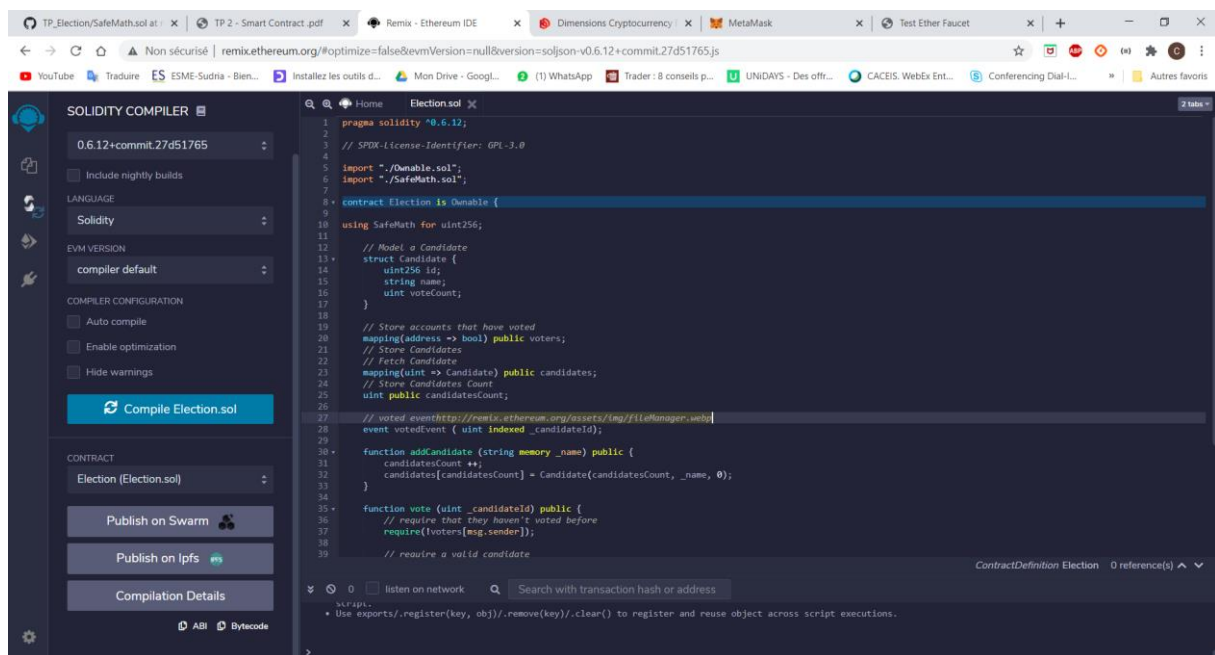
Receive
Sep 7 - From: 0x78c1...ee78 5 ETH 5 ETH



h)i))



k)



1)

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is visible. The 'ENVIRONMENT' is set to 'Injected Web3'. The 'ACCOUNT' is '0x542...34677 (4.998194 ei)'. The 'GAS LIMIT' is '3000000'. The 'VALUE' is '0 Wei'. The 'CONTRACT' is 'Election - browser/github/CeciliaCra...'. The 'Deploy' button is highlighted. Below it, there are options to 'Publish to IPFS' or 'At Address'. The 'Transactions recorded' section shows 1 transaction. The 'Deployed Contracts' section shows 'Currently you have no contract instances to interact with.'

The main editor displays the Solidity code for the 'Election' contract. The code includes a 'Candidate' struct, a 'voters' mapping, a 'candidates' mapping, and functions for 'addCandidate', 'vote', and 'votedEvent'.

```

12 // Model a Candidate
13 struct Candidate {
14     uint256 id;
15     string name;
16     uint voteCount;
17 }
18
19 // Store accounts that have voted
20 mapping(address => bool) public voters;
21 // Store Candidates
22 // Fetch Candidate
23 mapping(uint => Candidate) public candidates;
24 // Store Candidates Count
25 uint public candidatesCount;
26
27 // voted event http://remix.ethereum.org/assets/img/filofloraManager.webp
28 event votedEvent ( uint indexed _candidateId);
29
30 function addCandidate (string memory _name) public {
31     candidatesCount++;
32     candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
33 }
34
35 function vote (uint _candidateId) public {
36     // require that they haven't voted before
37     require(!voters[msg.sender]);
38
39     // require a valid candidate
40     require(_candidateId > 0 && _candidateId <= candidatesCount);
41
42     // record that voter has voted
43     voters[msg.sender] = true;
44
45     // update candidate's vote Count
46     candidates[_candidateId].voteCount++;
47
48     // trigger voted event
49     emit votedEvent (_candidateId);

```

The screenshot shows the MetaMask interface. The top bar indicates the network is 'Réseau de test Ropsten'. The main area displays the 'Déploiement de contrat' (Contract Deployment) process. A modal window titled 'Déploiement de contrat' is open, showing details of the transaction.

Déploiement de contrat

Détails

de: 0x542b817700c2a7... Nouveau contrat

Transaction

Nonce	2
Montant	0 ETH
Quantité Max. De Gaz (Unités)	553206
Essence Utilisée (Unités)	553206
Prix du gaz (GWEI)	1.5
Total	0.00083 ETH

Log D'activité

- Transaction créée avec une valeur de 0 ETH sur 10:48 on 9/7/2020. -2 ETH
- Transaction envoyée sur 10:48 on 9/7/2020. -2 ETH
- Transaction confirmée sur 10:48 on 9/7/2020.

The background shows the MetaMask account page for 'Account 1 / ETH' with a list of transactions:

- Déploiement de contrat (Sep 7 - remix.ethereum.org)
- Envoyer des ETH (Sep 7 - To: 0xc25a...100f)
- Envoyer des ETH (Sep 7 - To: 0x4415...d3ba)
- Receive (Sep 7 - From: 0x78c1...ee78)

The screenshot shows the Etherscan interface for a transaction on the Ropsten Testnet. The transaction hash is 0xcd078b263771b8e3a66d64112b1d3d74897cb395aff533cde6ec7ca65ff0e. The status is 'Success'. The block number is 8630230, with 79 block confirmations. The transaction occurred 8 minutes ago on September 7, 2020, at 08:48:29 AM UTC. It was sent from 0x542b817700c2a72e0de96667380b5d39734677 to a contract address 0x7957facb3fe2125171d07c3f7951fb5d724bd3e4. The value is 0 Ether (\$0.00) and the transaction fee is 0.000829809 Ether (\$0.000000).

m)

Les transactions fees ne sont pas les mêmes car elles dépendent du nombre de transactions sur le réseau à l'instant même.

Clé publique du smart contract : **0x7957FACB3FE2125171d07c3F7951FB5D724Bd3e4**

n)o)p)

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the 'Election AT 0x795..._BD3E4 (BLOCKCHAIN)' contract. It lists functions: 'addCandidate' (with parameter 'curus'), 'transferOwner...' (with parameter 'address newOwner'), 'vote' (with parameter 'uint256 _candidateId'), and 'candidates' (with parameter '1'). Below these, there are sections for 'candidatesCou...', 'owner', and 'voters'. The 'Low level interactions' section shows a 'CALLDATA' field and a 'Transact' button. The main editor displays the 'Election.sol' contract code, which includes a 'Candidate' struct, a 'voters' mapping, and functions for adding candidates and voting. The code is as follows:

```

contract Election is Ownable {
    using SafeMath for uint256;

    // Model a Candidate
    struct Candidate {
        uint256 id;
        string name;
        uint voteCount;
    }

    // Store accounts that have voted
    mapping(address => bool) public voters;
    // Store Candidates
    // Fetch Candidate
    mapping(uint => Candidate) public candidates;
    // Store Candidates Count
    uint public candidatesCount;

    // voted event http://remix.ethereum.org/assets/img/fileManager.webp
    event votedEvent ( uint indexed _candidateId );

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

    function vote (uint _candidateId) public {
        // require that they haven't voted before
        require(!voters[msg.sender]);

        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);

        // record that voter has voted
        voters[msg.sender] = true;

        // update candidate vote Count
        candidates[_candidateId].voteCount++;
    }
}

```

4.9972ETH

Add Candidate

Sep 7 · remix.ethereum.org

Déploiement de contrat

Sep 7 · remix.ethereum.org

Envoyer des ETH

Sep 7 · To: 0xc25a...100f

Envoyer des ETH

Sep 7 · To: 0x4415...d3ba

Receive

Sep 7 · From: 0x78c1...ee78

Receive

Sep 7 · From: 0x81b7...7647

Détails

de: 0x542b817700c2a7... Destinataire: 0x7957fa...

Transaction

Nonce3

Montant0 ETH

Quantité Max. De Gaz (Unités)88165

Essence Utilisée (Unités)86682

Prix du gaz (GWEI)1.5

Total0.00013 ETH

Log D'activité

Transaction créée avec une valeur de 0 ETH sur 11:02 on 9/7/2020.

Transaction envoyée sur 11:02 on 9/7/2020.

Transaction confirmée sur 11:02 on 9/7/2020.

-0 ETH

-0 ETH

-0 ETH

-1 ETH

-1 ETH

-2 ETH

-2 ETH

5 ETH

5 ETH

1 ETH

1 ETH

Transaction Details

Overview

State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x26aa8cb944e03ca71b8e274530927b020695c7c972a63255c87463db7c5b36e7

Status:

Success

Block:

8636360 31 Block Confirmations

Timestamp:

4 mins ago (Sep-07-2020 09:02:33 AM +UTC)

From:

0x542b817700c2a772e0de96667386b5d39734677

To:

Contract 0x7957fabc3fe2125171d07c3f7951fb5d724bd3e4

Value:

0 Ether (\$0.00)

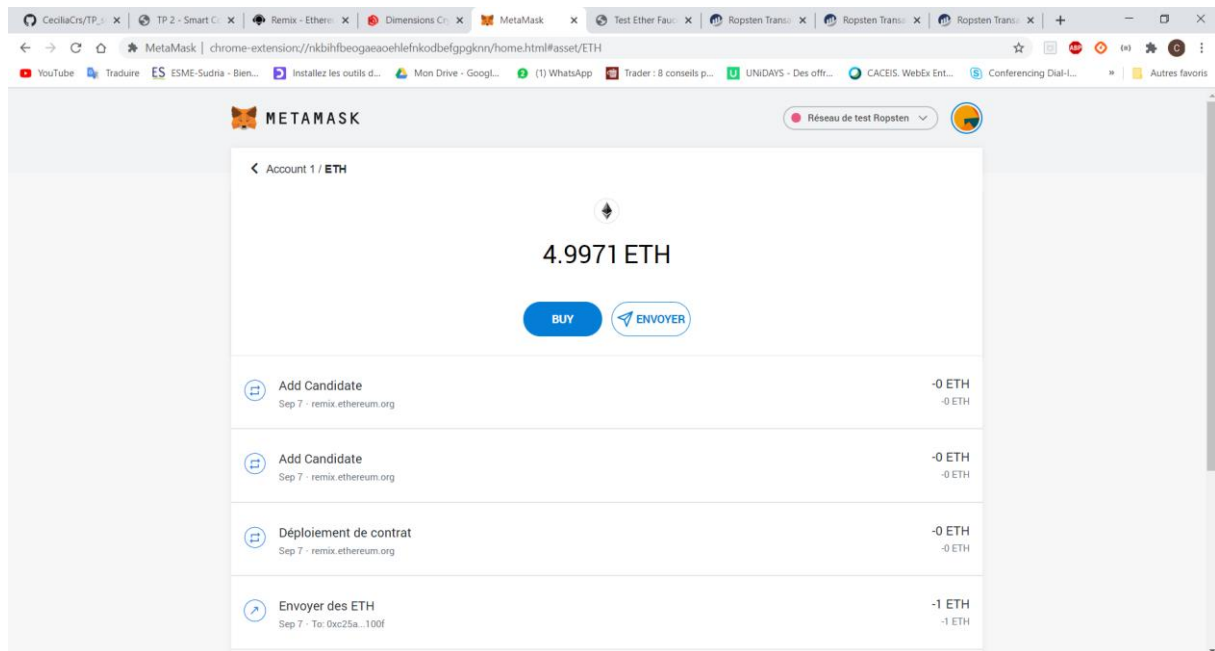
Transaction Fee:

0.000130023 Ether (\$0.000000)

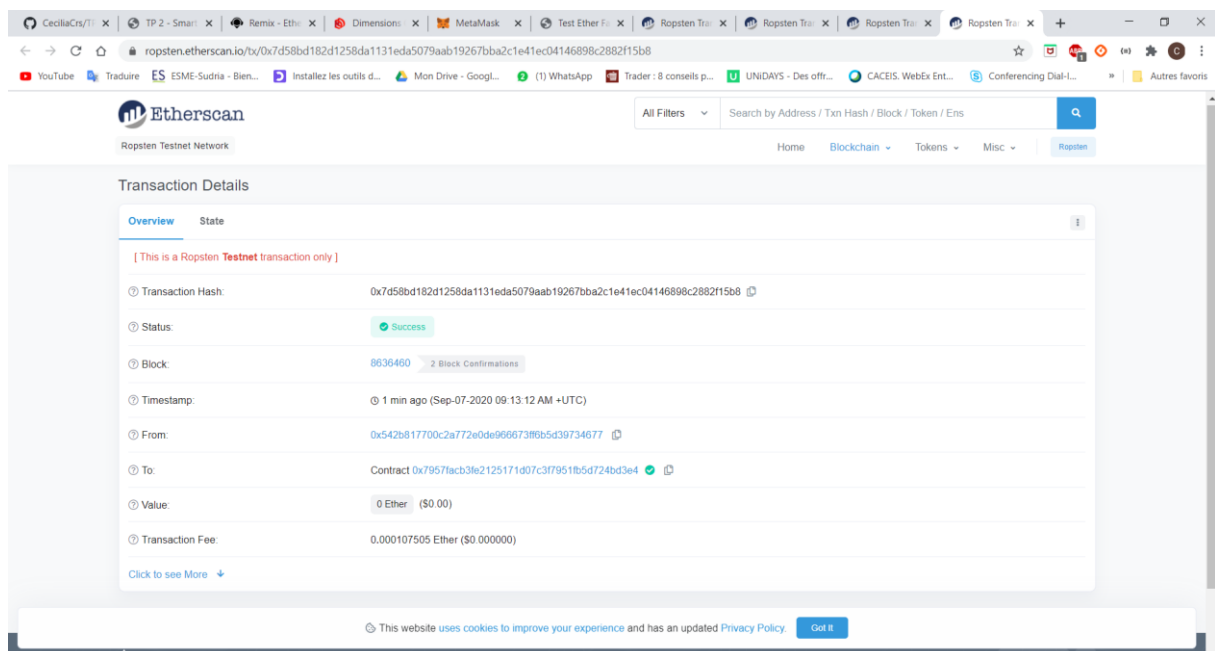
[Click to see More](#)

This website uses cookies to improve your experience and has an updated Privacy Policy. [Get it](#)

q)



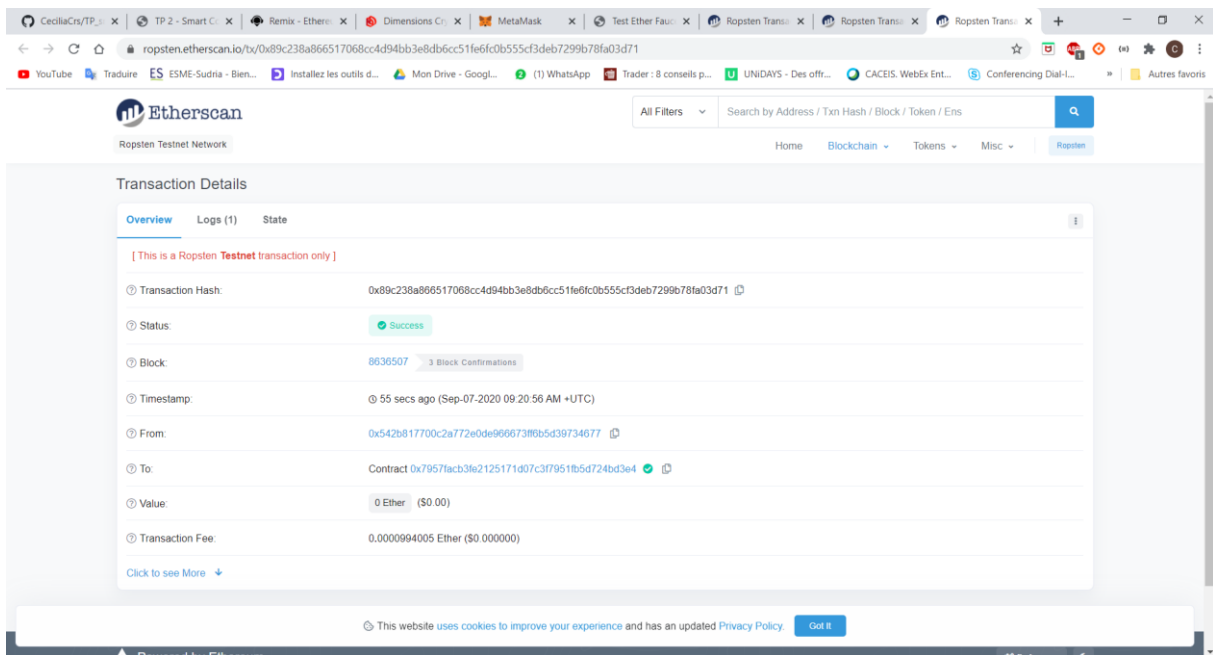
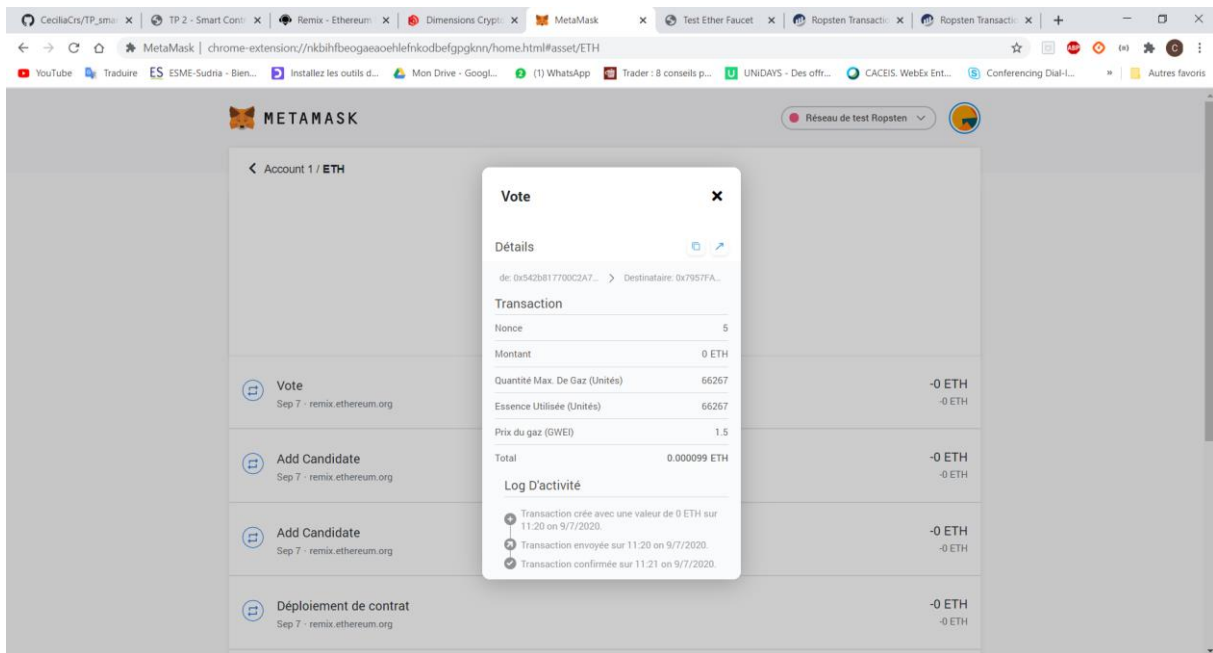
r)



s)

Adresse du propriétaire du contrat :
0x542b817700c2a772e0de966673ff6b5d39734677

t)



u)

The screenshot shows the Remix IDE interface. On the left sidebar, the 'addCandidate' function is selected, and its parameters are listed: 0: uint256 id 1, 1: string name carius, and 2: uint256 voteCount1 (highlighted with a red circle). The main editor displays the Solidity code for the 'Election' contract, which includes functions for adding candidates, voting, and managing the candidate list. The code is as follows:

```
8 contract Election is Ownable {
9
10     using SafeMath for uint256;
11
12     // Model a Candidate
13     struct Candidate {
14         uint256 id;
15         string name;
16         uint voteCount;
17     }
18
19     // Store accounts that have voted
20     mapping(address => bool) public voters;
21     // Store Candidates
22     mapping(uint => Candidate) public candidates;
23     // Store Candidates Count
24     uint public candidatesCount;
25
26     // voted event http://remix.ethereum.org/assets/img/fileManager.webp
27     event votedEvent ( uint indexed _candidateId);
28
29     function addCandidate(string memory _name) public {
30         candidatesCount++;
31         candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
32     }
33
34     function vote(uint _candidateId) public {
35         // require that they haven't voted before
36         require(!voters[msg.sender]);
37
38         // require a valid candidate
39         require(_candidateId > 0 && _candidateId <= candidatesCount);
40
41         // record that voter has voted
42         voters[msg.sender] = true;
43
44         // update candidate vote Count
45         candidates[_candidateId].voteCount++;
46     }
47
48     // trigger voted event
49     emit votedEvent(_candidateId);
50 }
51
52
```

v)

The screenshot shows the Remix IDE interface. On the left sidebar, the 'addCandidate' function is selected, and its parameters are listed: 0: uint256 id 1, 1: string name Kaing, and 2: uint256 voteCount 2. The main editor displays the Solidity code for the 'Election' contract, which includes functions for adding candidates, voting, and managing the candidate list. The code is as follows:

```
16     uint voteCount;
17 }
18
19 // Store accounts that have voted
20 mapping(address => bool) public voters;
21 // Store Candidates
22 mapping(uint => Candidate) public candidates;
23 // Store Candidates Count
24 uint public candidatesCount;
25
26 // voted event http://remix.ethereum.org/assets/img/fileManager.webp
27 event votedEvent ( uint indexed _candidateId);
28
29 function addCandidate(string memory _name) public {
30     candidatesCount++;
31     candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
32 }
33
34 function vote(uint _candidateId) public {
35     // require that they haven't voted before
36     require(!voters[msg.sender]);
37
38     // require a valid candidate
39     require(_candidateId > 0 && _candidateId <= candidatesCount);
40
41     // record that voter has voted
42     voters[msg.sender] = true;
43
44     // update candidate vote Count
45     candidates[_candidateId].voteCount++;
46
47     // trigger voted event
48     emit votedEvent(_candidateId);
49 }
50
51 }
52
```

w)

The screenshot shows the Remix IDE interface. On the left, the 'Election' contract is loaded, and the 'transferOwner' button is highlighted with a red circle. The main editor displays the Solidity code for the 'Election' contract, which includes functions for adding candidates, voting, and transferring ownership. The 'transferOwner' function is highlighted with a red circle.

x)y)

Dans la fonction addCandidate, on fait appel au modifier onlyOwner (présent dans Ownable.sol) qui permet de vérifier si on est bien le propriétaire du contrat.

The screenshot shows the Remix IDE interface. On the left, the 'Election' contract is loaded, and the 'addCandidate' button is highlighted with a red circle. The main editor displays the Solidity code for the 'Election' contract, which includes functions for adding candidates, voting, and transferring ownership. The 'addCandidate' function is highlighted with a red circle.

Deploy & Run Transactions interface for the Ownable.sol contract. The left sidebar shows the deployment configuration for the ELECTION AT 0x7E7...B69DB (BLOCKCHAIN). The main area displays the Solidity code for Ownable.sol, with a red circle highlighting the `modifier onlyOwner()` function.

```

7  * @dev The Ownable contract has an owner address, and provides basic authorization control
8  * functions, this simplifies the implementation of "user permissions".
9
10 contract Ownable {
11     address public owner;
12
13     event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
14
15     /**
16      * @dev The Ownable constructor sets the original 'owner' of the contract to the sender
17      * account.
18      */
19     constructor() public {
20         owner = msg.sender;
21     }
22
23     /**
24      * @dev Throws if called by any account other than the owner.
25      */
26     modifier onlyOwner() {
27         require(msg.sender == owner, "Not authorized operation");
28         _;
29     }
30
31     /**
32      * @dev Allows the current owner to transfer control of the contract to a newOwner.
33      * @param newOwner The address to transfer ownership to.
34      */
35     function transferOwnership(address newOwner) public onlyOwner {
36         require(newOwner != address(0), "Address shouldn't be zero");
37         emit OwnershipTransferred(owner, newOwner);
38         owner = newOwner;
39     }
40 }

```

The left sidebar shows the deployment configuration for the ELECTION AT 0x7E7...B69DB (BLOCKCHAIN). The main area displays the Solidity code for Ownable.sol, with a red circle highlighting the `modifier onlyOwner()` function.

Editing TP_Election.sol file in the Remix IDE. The code shows the implementation of the `onlyOwner` modifier and the `addCandidate` and `vote` functions.

```

17 }
18
19 // Store accounts that have voted
20 mapping(address => bool) public voters;
21
22 // Store Candidates
23 mapping(uint => Candidate) public candidates;
24 // Store Candidates Count
25 uint public candidatesCount;
26
27 // voted event
28 event votedEvent ( uint indexed _candidateId);
29
30 function addCandidate (string memory _name) public onlyOwner {
31     candidatesCount++;
32     candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
33 }
34
35 function vote (uint _candidateId) public {
36     // require that they haven't voted before
37     require(!voters[msg.sender]);
38
39     // require a valid candidate
40     require(_candidateId > 0 && _candidateId <= candidatesCount);
41
42     // record that voter has voted
43     voters[msg.sender] = true;
44
45     // update candidate vote Count
46     candidates[_candidateId].voteCount++;
47
48     // trigger voted event
49     emit votedEvent (_candidateId);

```