Homework of Lecture 6

> Lecture 6 - Homework & Resources
>
> > Homework is done by a group.

1. Answer the following questions in a few sentences using your own words.

1.a) Why does Ethereum price EVM instructions in GAS instead of using ETH directly?

> *GAS refers to the computational power required from miners to validate the transaction or smart contract. ETH is the currency used to compensate miners for the GAS required to do the computational work. GAS consumption values are relatively static, while the price of ETH is dynamic and changes overtime.*
>
> *Charging instructions in GAS thus makes it easier for miners to measure the* ***cost-benefits of a computation*** *and planning their work.*

1.b) What is the goal of the gas sponsorship mechanism introduced in Conflux?

> *Transactions costs on the blockchain are traditionally paid by the user, who benefits from the blockchain network feature.*
>
> *However, as with the case in any other economy, costs are always* ***pain points*** *and discourages adoption. Adoption is critical for building an ecosystem which value increases from network externalities, which precisely the case with most blockchain networks. In addition,* ***blockchain is not necessarily used to facilitate transactions****, so there is the possibility that users of the network do not have any deposited amount with which to pay the transaction fee.*
>
> *The gas sponsorship mechanism therefore 1) removes a barrier to adoption of the network by new users and 2) increases the application scenarios / flexibility of the smart contract.*

1.c) What steps should a developer take so that users of their smart contract do not have to pay for gas?

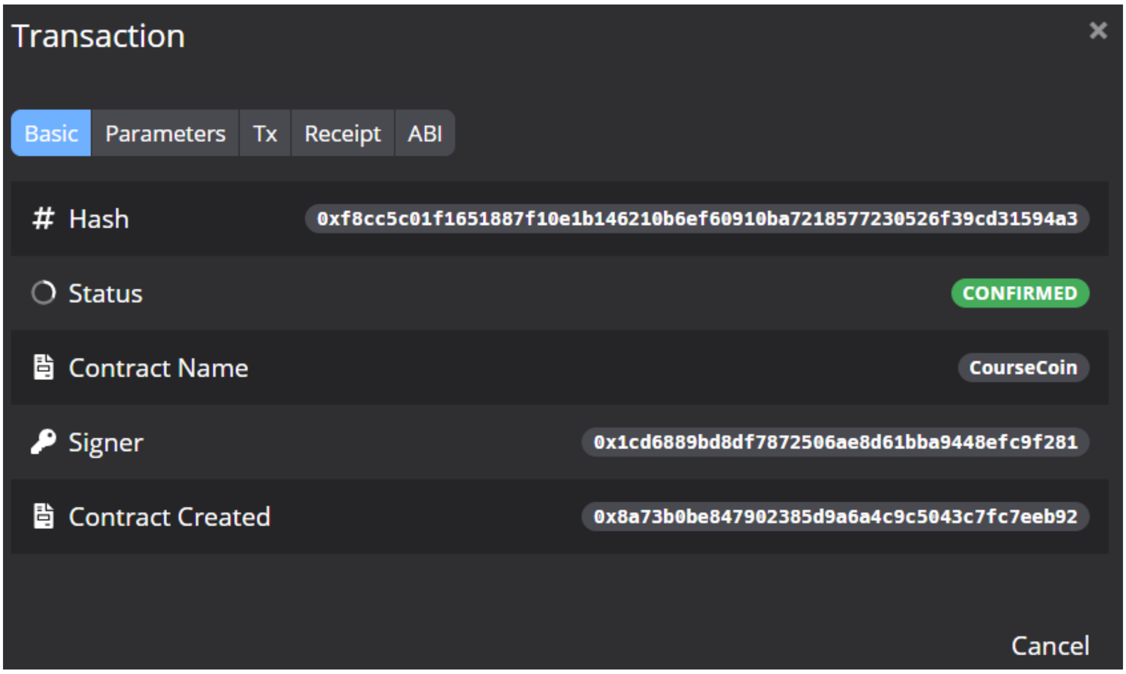
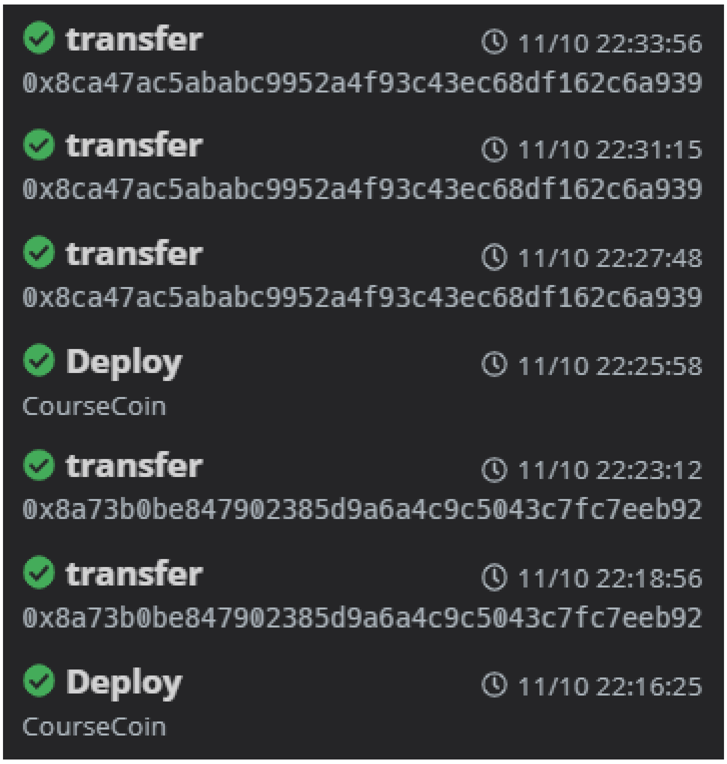
> *Implement the Gas Sponsorship Mechanism: allow anyone willing to pay to become a "Sponsor", donating funds to any contract to pay for the transactions of other whitelisted users (usually new ones with 0 account balance).*
>
> *Several parameters to include:*
>
> 1. *Sponsor - records the account providing sponsorship funds for the smart contract*
> 2. *Sponsor balance - records the current balance of the sponsorship funds for the smart contract*
> 3. *Sponsor limit per transaction - sets an upper limit of funds for individual transactions*
> 4. *Whitelist - records, or limits, the accounts that can be sponsored/funded by the smart contract*

+1. Gas cost of token transfers [OPTIONAL]

Download lecture-5's CourseToken project (github.com/Thegaram/cfx-uma-resources/raw/master/cfx-lecture-5.zip 2) or use your own version.

Send some tokens to a new address. How much were you charged for this transfer? Try to explain why.

How does the fee change if you send some more tokens to the same address again?

> Create a new "CourseCoin" contract:
>
> 
>
>
> \# Hash 0xf8cc5c01f1651887f10e1b146210b6ef60910ba7218577230526f39cd31594a3
>
> Status CONFIRMED
>
> Contract Name CourseCoin
>
> Signer 0x1cd6889bd8df7872506ae8d61bba9448efc9f281
>
> Contract Created 0x8a73b0be847902385d9a6a4c9c5043c7fc7eeb92
>
> **Send some tokens to a new address**:
>
> 
>
> ✓ **transfer** 11/10 22:33:56
> 0x8ca47ac5ababc9952a4f93c43ec68df162c6a939
>
> ✓ **transfer** 11/10 22:31:15
> 0x8ca47ac5ababc9952a4f93c43ec68df162c6a939
>
> ✓ **transfer** 11/10 22:27:48
> 0x8ca47ac5ababc9952a4f93c43ec68df162c6a939
>
> ✓ **Deploy** 11/10 22:25:58
> CourseCoin
>
> ✓ **transfer** 11/10 22:23:12
> 0x8a73b0be847902385d9a6a4c9c5043c7fc7eeb92
>
> ✓ **transfer** 11/10 22:18:56
> 0x8a73b0be847902385d9a6a4c9c5043c7fc7eeb92
>
> ✓ **Deploy** 11/10 22:16:25
> CourseCoin
>
> - **How much were you charged for this transfer?**
>   - sent `10`, "gasUsed": "**35819**" with gas limit "1000000", and "gasPrice": 100
>   - ~~Before Vel has CFX: `34363.0111445711547907 CFX`~~
>   - ~~SUDDENLY I realize that "Vel" is mining to gain more CFXs. So it won't work this way.~~
>   - Created another new "CourseCoin" contract with "Calvin"
>   - Before Calvin has CFX: `9997.756835937425` **CFX**

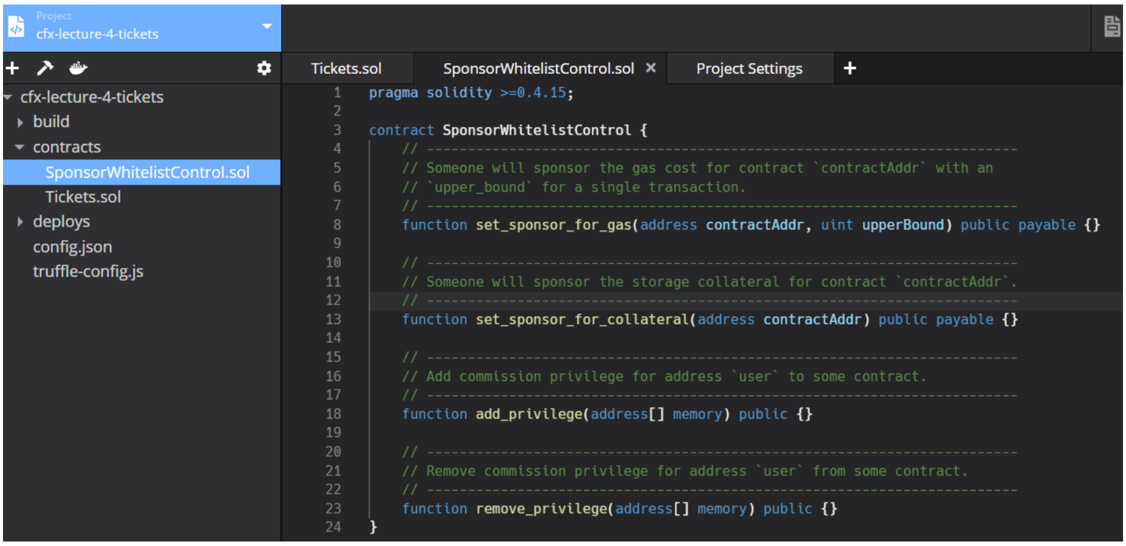
>   - After Calvin has CFX: `9997.69433593735` **CFX**
>   - So, total cost is: `0.06250000007457857` **CFX**
>
> **Try to explain why.**
>
> We have the calculation form for this as follows:
>
> > *so what you will pay is*
> >
> > **(*gas_used* x *gas_price*) + (*1/16 CFX* x *storage_collateralized*)**
> >
> > *... minus the refund of at most* ***1/4*** **x** ***gas_used*** **x** ***gas_price***
>
> - *gas_used* x *gas_price* = `35819 * 100` = 3581900 Drips = 3.5819e-12 CFX (**In theory**)
>   - But according to the refund policy:
>   - *gas_used* x *gas_price* = `3/4 * 1000000 * 100` = 75000000 Drips = 7.5e-11 CFX
> - so we can deduce that:
>   - *1/16 CFX* x *storage_collateralized* = `0.06250000007457857` **CFX** - 7.5e-11 CFX ≈ 0.0625 CFX
>   - which also implies: *storage_collateralized* ≈ 1
>
> > But still, the numbers doesn't exactly match which is still remaining mystery.
>
> **How does the fee change if you send some more tokens to the same address again?**
>
> - sent `1234`, gasUsed": "**35883**"
> - Calvin now has CFX: `9997.694335937275` **CFX**
> - total cost is: `0.000000000075` **CFX**
> - It becomes **significantly low**. I think it's because the first transaction include data storage initialization which is expensive (***1/16 CFX*** **x** ***storage_collateralized***).

+1. Sponsored ticket sale [OPTIONAL]

Download lecture-4's Tickets project (github.com/Thegaram/cfx-uma-resources/raw/master/cfx-lecture-4-tickets.zip) or use your own version.

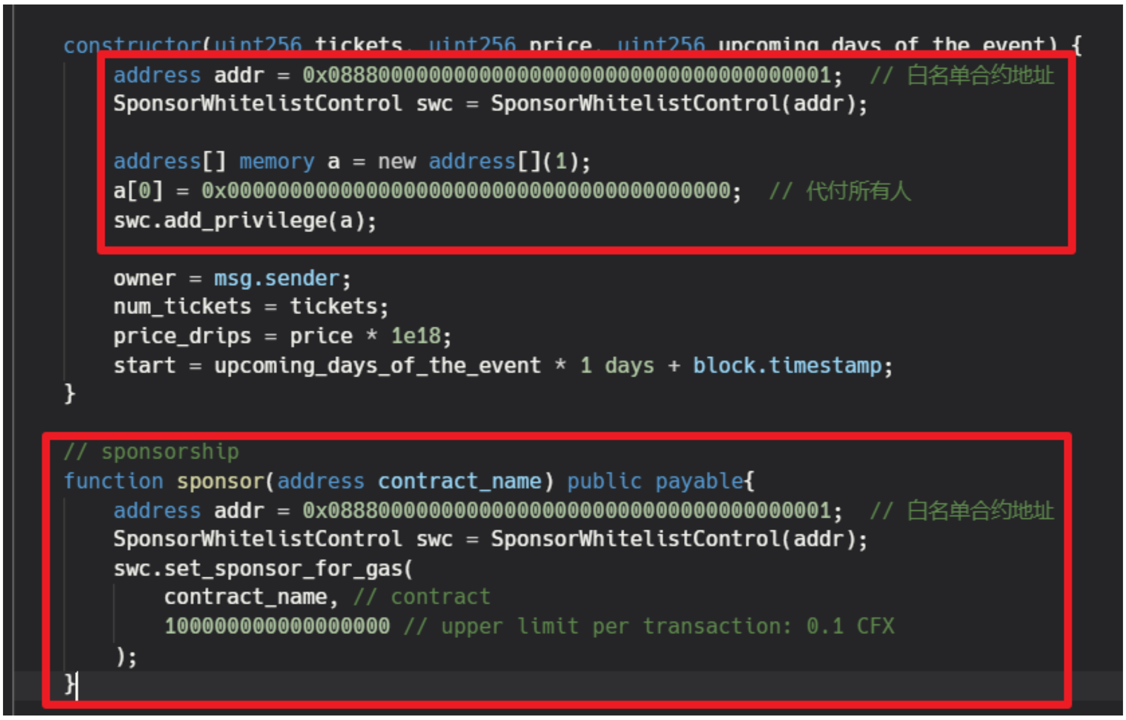
Change the contract so that people buying tickets through it **do not have to pay any transaction fee**.

> First, trying to find `SponsorWhitelistControl.sol`.
>
> > The `import "github.com/Conflux-Chain/conflux-rust/blob/master/internal_contract/contracts/SponsorWhitelistControl.sol";` doesn't seem working now. Bad news.
>
> But going the website and copying the code is avaliable. And the version is changed. Function Callings are also different.
>
> So I reimplement the `SponsorWhitelistControl.sol` to fit the demand.

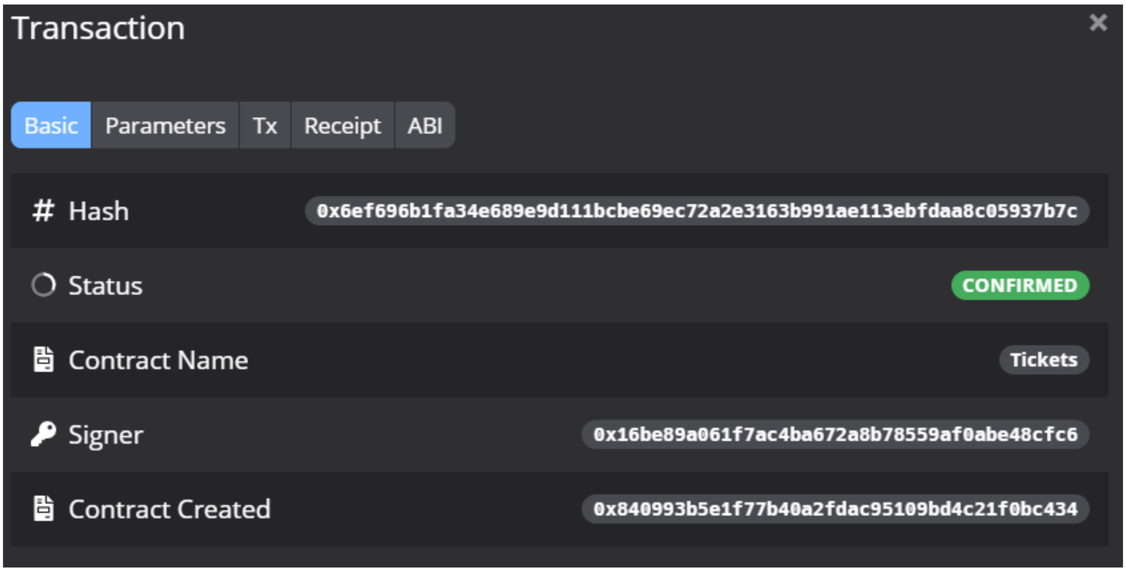


```
pragma solidity >=0.4.15;

contract SponsorWhitelistControl {
    // ------------------------------------------------------------------------
    // Someone will sponsor the gas cost for contract `contractAddr` with an
    // `upper_bound` for a single transaction.
    // ------------------------------------------------------------------------
    function set_sponsor_for_gas(address contractAddr, uint upperBound) public payable {}

    // ------------------------------------------------------------------------
    // Someone will sponsor the storage collateral for contract `contractAddr`.
    // ------------------------------------------------------------------------
    function set_sponsor_for_collateral(address contractAddr) public payable {}

    // ------------------------------------------------------------------------
    // Add commission privilege for address `user` to some contract.
    // ------------------------------------------------------------------------
    function add_privilege(address[] memory) public {}

    // ------------------------------------------------------------------------
    // Remove commission privilege for address `user` from some contract.
    // ------------------------------------------------------------------------
    function remove_privilege(address[] memory) public {}
}
```

Now we deploy the ticket contract, and there are few changes in the code:



```
constructor(uint256 tickets, uint256 price, uint256 upcoming_days_of_the_event) {
    address addr = 0x0888000000000000000000000000000000000001;  // 白名单合约地址
    SponsorWhitelistControl swc = SponsorWhitelistControl(addr);

    address[] memory a = new address[](1);
    a[0] = 0x0000000000000000000000000000000000000000;  // 代付所有人
    swc.add_privilege(a);

    owner = msg.sender;
    num_tickets = tickets;
    price_drips = price * 1e18;
    start = upcoming_days_of_the_event * 1 days + block.timestamp;
}

// sponsorship
function sponsor(address contract_name) public payable{
    address addr = 0x0888000000000000000000000000000000000001;  // 白名单合约地址
    SponsorWhitelistControl swc = SponsorWhitelistControl(addr);
    swc.set_sponsor_for_gas(
        contract_name, // contract
        100000000000000000 // upper limit per transaction: 0.1 CFX
    );
}
```



Hash 0x6ef696b1fa34e689e9d111bcbe69ec72a2e3163b991ae113ebfdaa8c05937b7c

Status CONFIRMED

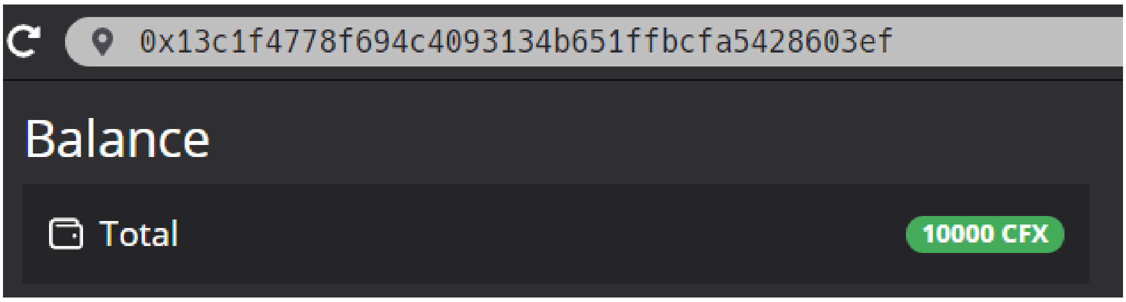
Contract Name Tickets

Signer 0x16be89a061f7ac4ba672a8b78559af0abe48cfc6

Contract Created 0x840993b5e1f77b40a2fdac95109bd4c21f0bc434

> The ticket is priced as 15 CFX for each.

"Leon" has 10000 CFX at first:

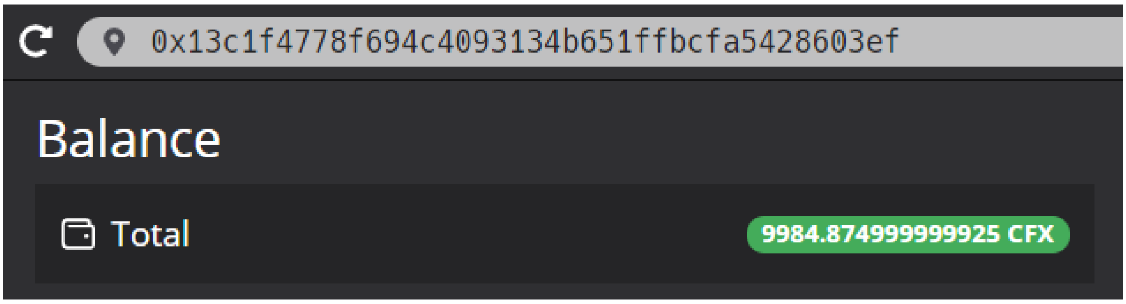


0x13c1f4778f694c4093134b651ffbcfa5428603ef

Balance

Total 10000 CFX

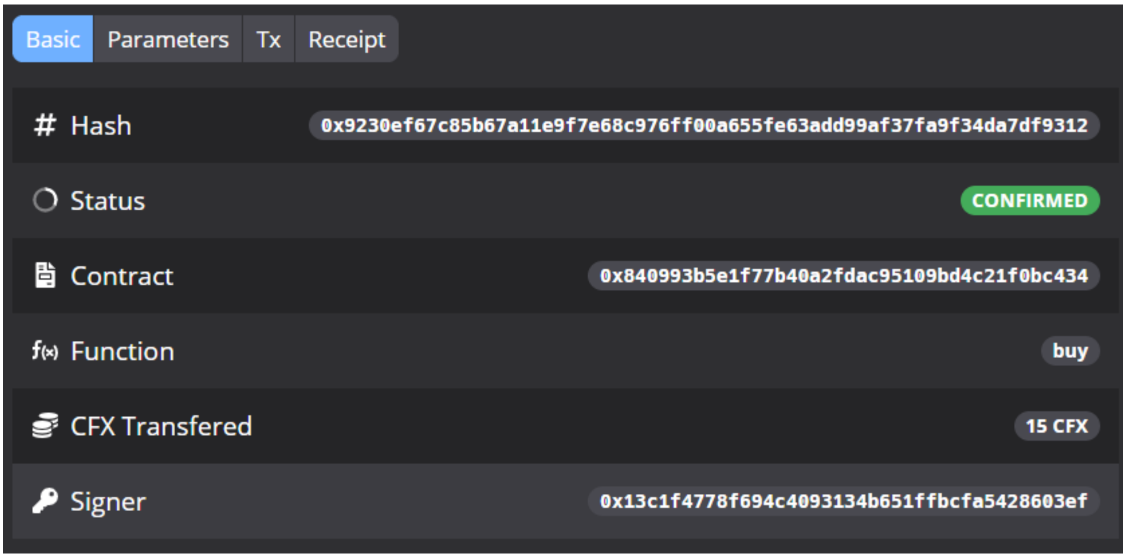
Leon buys 1 ticket:



0x13c1f4778f694c4093134b651ffbcfa5428603ef

Balance

Total 9984.874999999925 CFX



Hash 0x9230ef67c85b67a11e9f7e68c976ff00a655fe63add99af37fa9f34da7df9312

Status CONFIRMED

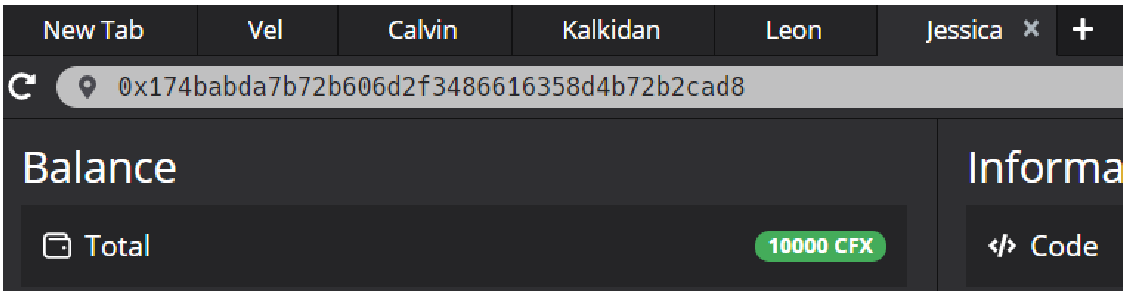
Contract 0x840993b5e1f77b40a2fdac95109bd4c21f0bc434

Function buy

CFX Transfered 15 CFX

Signer 0x13c1f4778f694c4093134b651ffbcfa5428603ef

Let's say Jessica wants to sponsor the contract:



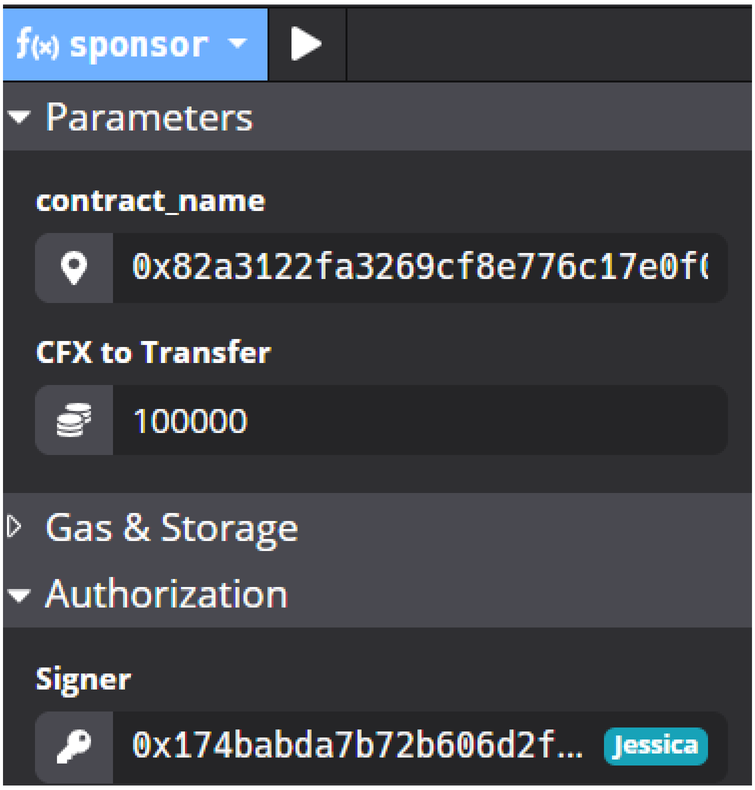
0x174babda7b72b606d2f3486616358d4b72b2cad8

Balance

Total 10000 CFX

Informa

Code



Parameters

contract_name: 0x82a3122fa3269cf8e776c17e0f(

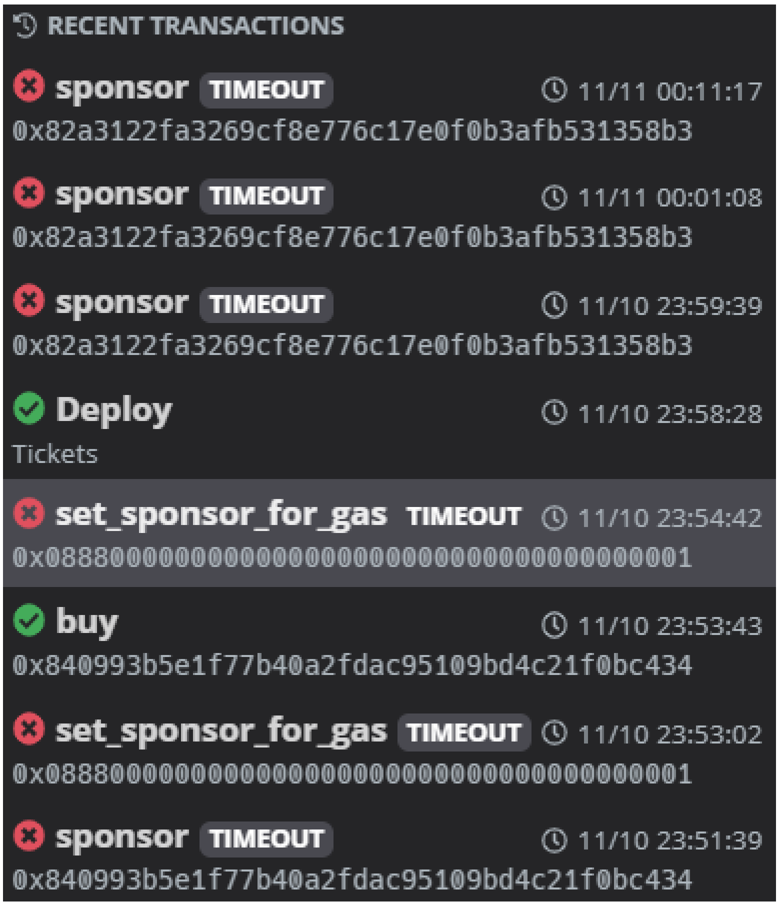
CFX to Transfer: 100000

Gas & Storage

Authorization

Signer: 0x174babda7b72b606d2f... Jessica

running into a serious trouble...



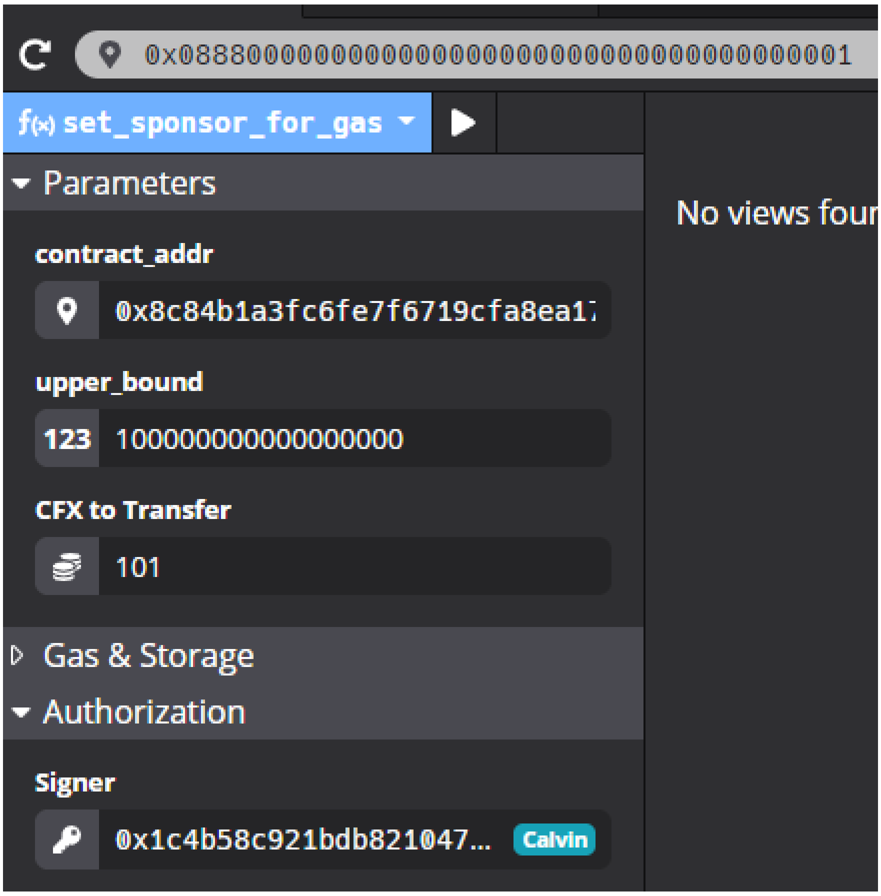
RECENT TRANSACTIONS

```
✗ sponsor TIMEOUT                      11/11 00:11:17
0x82a3122fa3269cf8e776c17e0f0b3afb531358b3
✗ sponsor TIMEOUT                      11/11 00:01:08
0x82a3122fa3269cf8e776c17e0f0b3afb531358b3
✗ sponsor TIMEOUT                      11/10 23:59:39
0x82a3122fa3269cf8e776c17e0f0b3afb531358b3
✓ Deploy                               11/10 23:58:28
Tickets
✗ set_sponsor_for_gas TIMEOUT          11/10 23:54:42
0x0888000000000000000000000000000000000001
✓ buy                                  11/10 23:53:43
0x840993b5e1f77b40a2fdac95109bd4c21f0bc434
✗ set_sponsor_for_gas TIMEOUT          11/10 23:53:02
0x0888000000000000000000000000000000000001
✗ sponsor TIMEOUT                      11/10 23:51:39
0x840993b5e1f77b40a2fdac95109bd4c21f0bc434
```

All the sponsor transactions are timeout and reverted, even the ones I directly send from the `0x0888000000000000000000000000000000000001` panel.

After some parameter adjustments, it worked.

Calvin sponsored 101 CFX to the contract.


0x0888000000000000000000000000000000000001

contract_addr: 0x8c84b1a3fc6fe7f6719cfa8ea17

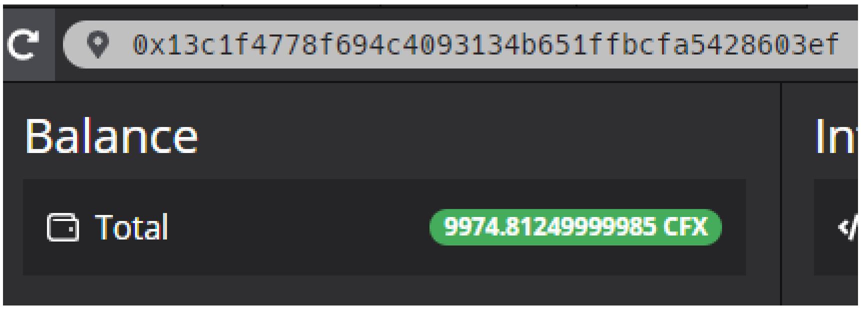
upper_bound: 123 1000000000000000000

CFX to Transfer: 101

Signer: 0x1c4b58c921bdb821047... Calvin

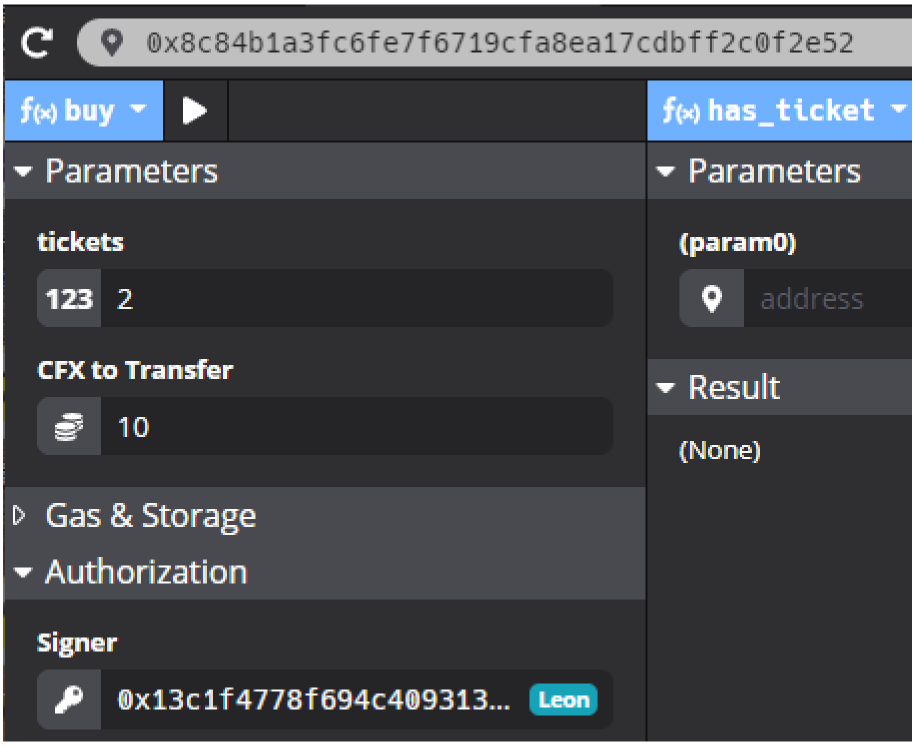
Now we can try to buy some more tickets without any charges:

Leon has: **9974.81249999985 CFX**


0x13c1f4778f694c4093134b651ffbcfa5428603ef

Total 9974.81249999985 CFX

Leon buys 2 tickets: (Each ticket is 5 CFX)



0x8c84b1a3fc6fe7f6719cfa8ea17cdbff2c0f2e52

tickets: 2

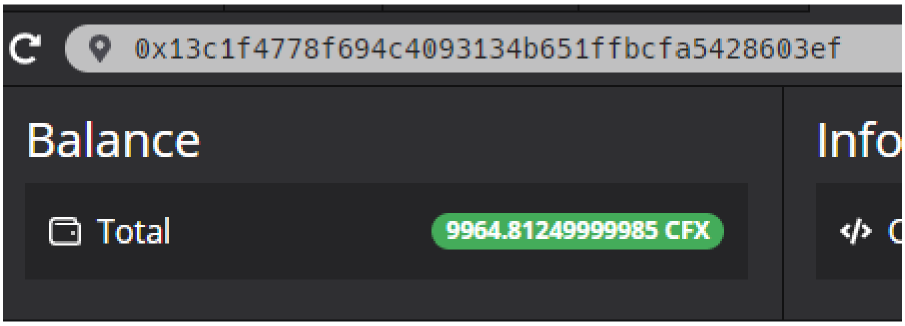
CFX to Transfer: 10

Signer: 0x13c1f4778f694c409313... Leon

(param0): address

Result: (None)

Leon has: 9974.81249999985 CFX - 10 CFX = **9964.81249999985 CFX**



0x13c1f4778f694c4093134b651ffbcfa5428603ef

Total 9964.81249999985 CFX

Indeed there is no transaction fee for Leon! We made it!