## Exercise 1

In Ethereum: what happens when a transaction with value V > 0 is sent to a contract address?

a) Nothing, the value field in a transaction is ignored by smart contracts

b) If the transaction invokes a payable function, the value of the contract balance in ETH is increased of V

c) The value of the balance in ETH of the account owning the smart contract is increased of V

Answer: B

TRUE OR FALSE?

1) The “Data” field in an Ethereum transaction sent by an EOA to another EOA is ignored by the Ethereum protocol (True)

2) In an Ethereum transaction there is no fixed transaction fee, but fees are only paid in the form of “gas” to pay for the execution of transactions (True)

3) The Proof-of-Work in Ethereum was designed in such a way that its complexity is determined mainly by the use of the memory while mining (False)

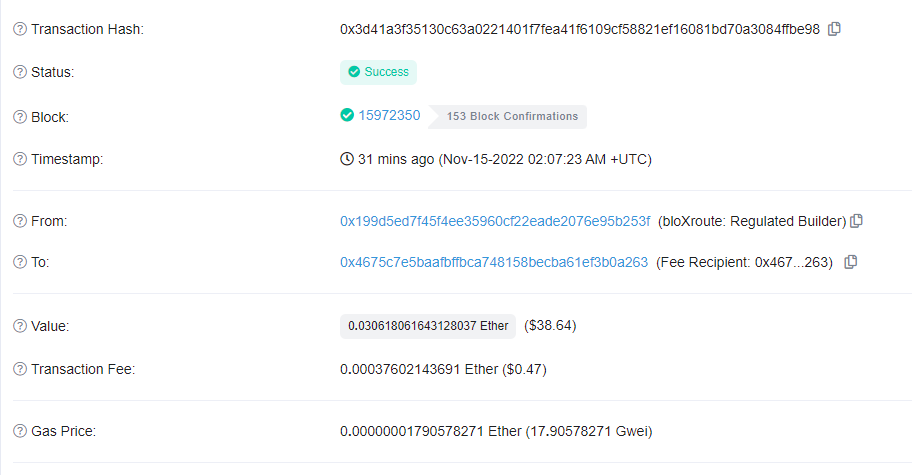
4) A node who has assembled the next block in Ethereum receives the sum of all the gas in the “gas limit” field of all transactions included in the block (False)

## Exercise 2

Using any online Ethereum blockchain explorer, find transactions issued on the main Ethereum network with the following characteristics:

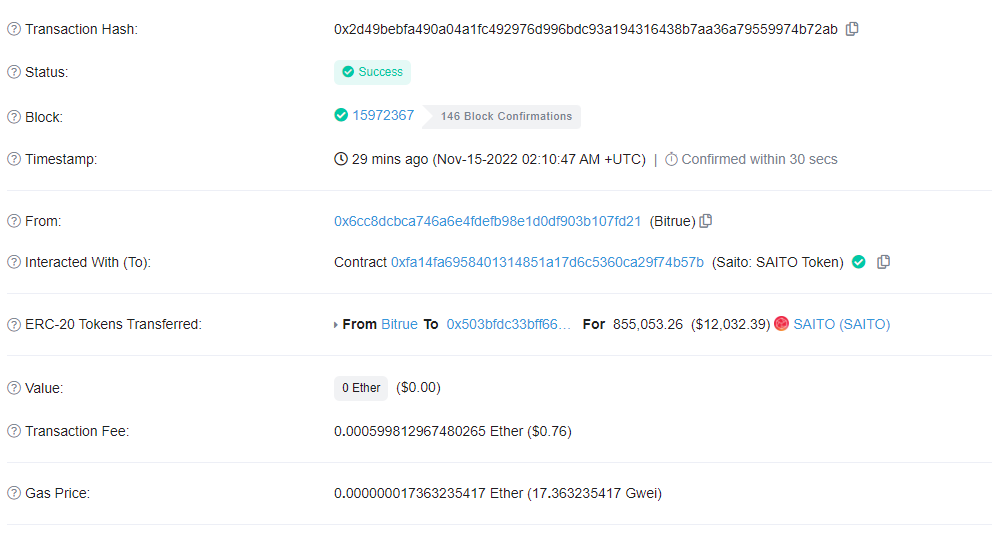
a) One transaction transferring ETH between two EOAs

<https://etherscan.io/tx/0x3d41a3f35130c63a0221401f7fea41f6109cf58821ef16081bd70a3084ffbe98>



b) One transaction invoking a smart contract. Indicate also which is the contract that was invoked and which is the function of this contract that was invoked by this transaction.

Transaction: <https://etherscan.io/tx/0x2d49bebfa490a04a1fc492976d996bdc93a194316438b7aa36a79559974b72ab>



Contract:

<https://etherscan.io/address/0xfa14fa6958401314851a17d6c5360ca29f74b57b>

Graphical user interface, text, chat or text message

Description automatically generated

Function:

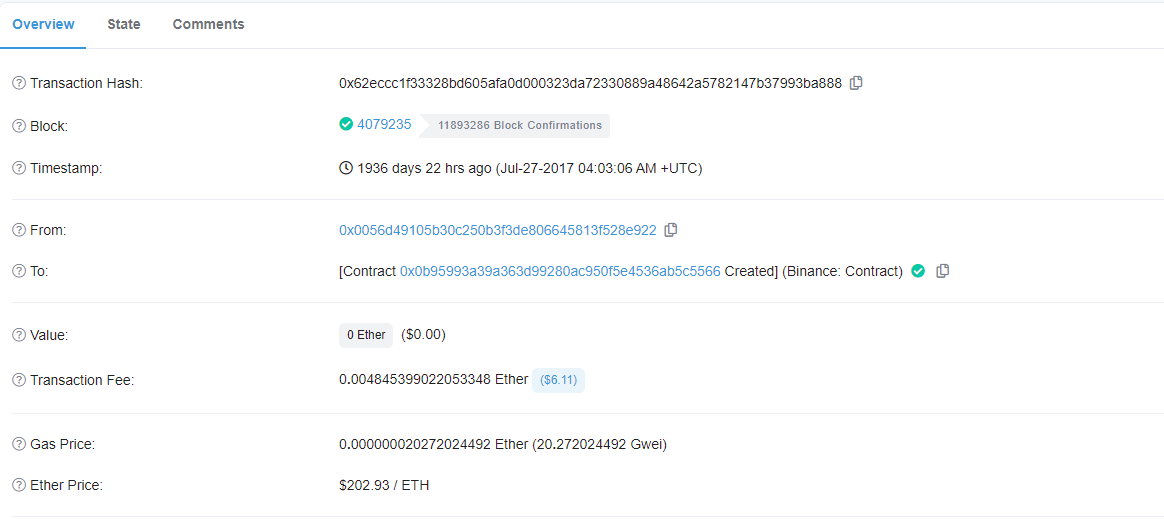
Transfer(address recipient, uint256 amount)

Graphical user interface, text, application

Description automatically generated

c) One transaction deploying a new smart contract.

<https://etherscan.io/tx/0x62eccc1f33328bd605afa0d000323da72330889a48642a5782147b37993ba888>



d) One transaction in which more than 80% of the gas was used.

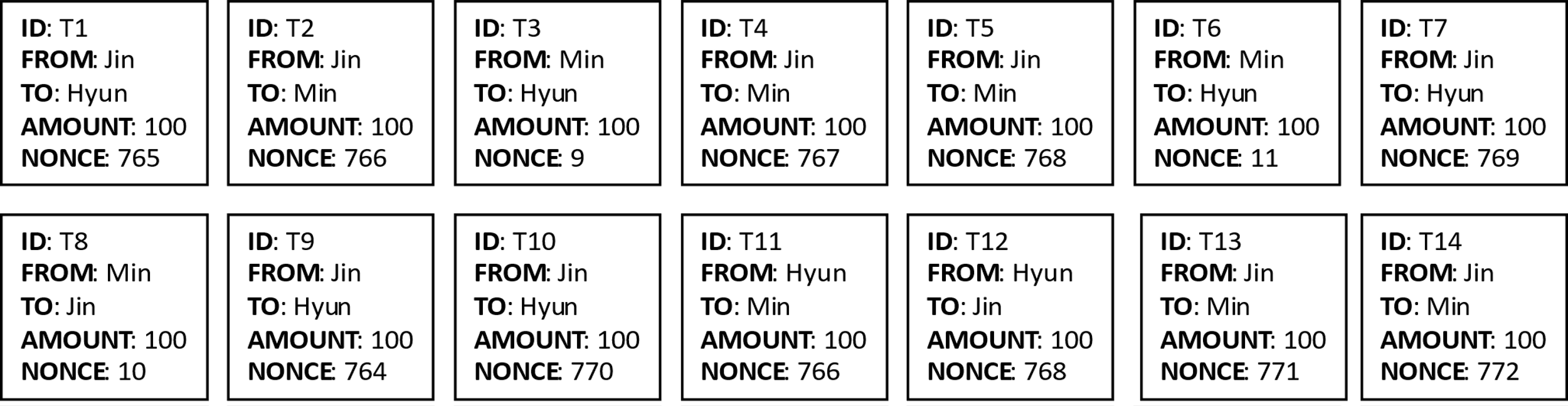
<https://etherscan.io/tx/0x5dee7f4b15cf15b9a50df1a191b60c6c4251efe40a1d685da562dd6b82e9ea22>

Graphical user interface, text, application, email

Description automatically generated

## Exercise 3

Consider the following Ethereum transactions, received by a mining node in the same order as their numbering (T1 received before T2, etc.). Assume that the nonce of the latest transactions received before T1 by this node from Jin, Min and Hyun are 763, 8, and 765, respectively.



In which order will the transactions be processed by the node to be included in a block? (1)

There is one transaction that will not be processed by the node. Which one? What must happen in order for this transaction to be processed? (2)

1. T3, T8, T6, T9, T1, T2, T4, T5, T7, T10, T13, T14
2. T11 will not be processed because it has a nonce equal to 766, while the last transaction by the originator has nonce equal to 764. For the T11 to be processed, another transaction, say T15, should be issued by Hyun with a nonce equal to 765.

## Exercise 4

Consider a node A in Hyperledger Fabric that has the following copy of the world state:

|  |  |  |
| --- | --- | --- |
| Business Object | Attribute | Value |
| R1 | Status | Sent |
| R2 | Status | Approved |
| R3 | Status | Rejected |
| R4 | Status | Sent |

A then receives a new block containing the following transactions:

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction | Chaincode Function | Input Set | Output Set |
| T1 | Approve(R1) | R1{Status: Sent} | R1{Status: Approved} |
| T2 | Reject(R4) | R4{Status: Sent} | R4{Status: Rejected} |
| T3 | Approve(R3) | R3{Status: Sent} | R4{Status: Rejected} |
| T4 | Create(R5) | N/A | R5{Status: Sent} |
| T5 | Reject(R2) | R2{Status: Sent} | R2{Status: Rejected} |
| T6 | Reissue(R3) | R3{Status: Rejected} | R3{Status: Sent} |
| T7 | Create(R6) | N/A | R6{Status: Sent} |

Indicate which transactions should the node consider valid (and therefore use them to update its world state), which ones should it consider invalid (and then discarded), and how the world state would look like after the block has been processed.

Valid transactions: T1, T2, T4, T6, T7

Invalid transactions: T3 and T5

Updated world state:

|  |  |  |
| --- | --- | --- |
| Business Object | Attribute | Value |
| R1 | Status | Approved |
| R2 | Status | Approved |
| R3 | Status | Sent |
| R4 | Status | Rejected |
| R5 | Status | Sent |
| R6 | Status | Sent |