© Journal of the Technical University - Sofia
Plovdiv branch, Bulgaria
"Fundamental Sciences and Applications"
Vol. 25, 2019

An approach of Crypto-token for Smart Contract based on Ethereum Blockchain

V Aleksieva[1], A Huliyan[2], H Valchanov[3]

[1]Assoc. Professor, Department of Computer Science, Technical University of Varna, Varna, Bulgaria
[2] Associate Professional Programmer Analyst, DXC Technology Bulgaria Ltd., Varna, Bulgaria
[3]Assoc. Professor, Head of Department of Computer Science, Department of Computer Science, Technical University of Varna, Varna, Bulgaria

E-mail: valeksieva@tu-varna.bg

Abstract. Nowadays smart contracts apply in management, bank system, insurance, estate, IoT, and others, because they perform credible and trackable irreversible transactions without third parties. The proposed paper presents a solution for the creation of a decentralized token for the implementation of a smart contract based on Ethereum block-chain. A web based interface has been created for Initial Coin Offering (ICO). In the experimental environment the research was carried out for various scenarios. The results are presented.

1. Introduction

Nowadays, there are more than 1300 crypto-currencies. Although they have a small sector of global finance, they are becoming more and more widely available as a means of payment. New 8-10 crypto-currencies appear daily [1]. Since 11.12.2017 the Chicago Stock Exchange trades in financial instruments (futures) based on Bitcoins [2]. Since Q3 2017 the platforms of Cboe Global Markets Inc. [3] and CME Group Inc. [4] have been offered Bitcoin's futures.

Block-chain technology is a kind of public registry of crypto-accounts and crypto-transactions because it allows the creation of decentralized systems in which information is protected from malicious interference by the consensus principle. The possessed crypto-currency can serve to buy goods, pay for services, acquire rights. In [5] is presented a taxonomy, which comprises six block-chain application areas that are classified across eight technical dimensions.

The Smart Contract is a contract that is concluded using block-chain technology. In [6] it is examined the differences between new, smart-contract-based private ordering regime and the fundamental components of copyright law, such as exceptions and limitations, the doctrine of exhaustion, restrictions on formalities, the public domain and fair remuneration.

Smart Contracts are self-enforcing pieces of software, which reside and run over a hosting block-chain. However, writing trustworthy and safe smart contracts can be tremendously challenging because of the complicated semantics of underlying domain-specific languages and its testability. In [7] it is inspected what are the possible legal and technical disadvantages of logic-based smart contracts in light of common activities featuring ordinary contracts, and how to use such logic-based smart contracts in combination with blockchain systems. However, procedural languages are commonly used to program smart contracts in blockchain system. When it involves security of smart

 Online ISSN 2603-459X

contracts, developers embracing the ability to write the contracts should be capable of testing their code, for diagnosing security vulnerabilities, before deploying them to the immutable environments on blockchains. However, there are only a handful of security testing tools for smart contracts. In [8] the authors evaluate possible smart contract vulnerabilities in security and privacy before widespread implementation.

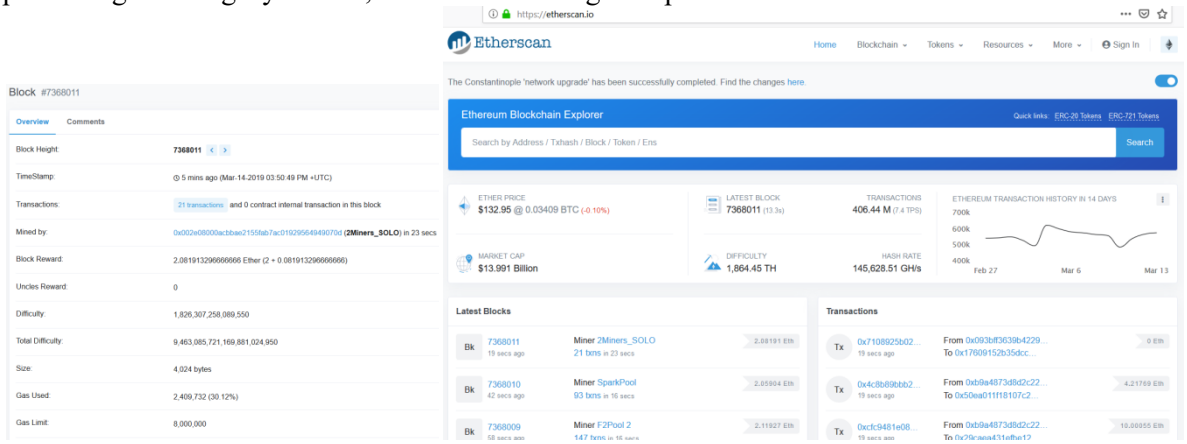
Crypto-currencies allow the execution of any transaction, as long as the paying agent's account has a sufficient balance. They are implemented through a Peer-to-Peer network with block-chain technology supported by the participants. Each participant has a copy of a block-chain of the network, and confirmation of the transactions is carried out by special network participants called "miners".

2. Smart-contract

2.1. Ethereum block-chain

Ethereum [9] is the first platform for creation of smart contracts and decentralized applications. In 2014it has been officially announced, with the Initial Coin Offering (ICO).The Ethereum platform became reality on 30.07.2015 with 11.5 million coins released in circulation. The consensus protocol that Ethereum currently uses is Proof-of-Work (PoW), but it is expected to increase the number of transactions to 30000 per second with implementation of new different protocols as Proof-of-Stake or Sharding protocol. In Ethereum, the exchange unit is called "Ether", and besides it there is an auxiliary unit called "GAS" that is used for creation and operation of smart contracts and applications, which work on a bloc-chain.

The existing transactions are confirmed or registered in a chain called block-chain. Data is grouped in blocks and arranged one after the other to form one big string (Fig.1a). It contains all the information about purchasing and selling on the web [10]. The data is visible to the public and available to be reviewed by anyone willing to do that (Fig.1b). If there is an attempt to change or edit any entry in a block, communication is automatically rejected. The system is so well developed that many sectors of public life are ready to use block-chain to verify and register data. This method allows preserving the integrity of data, thus counterfeiting is impossible.



a b

Figure 1. One block from Ethereum block-chain (a), Ethereum block-chain (b).

2.2. Smart contract

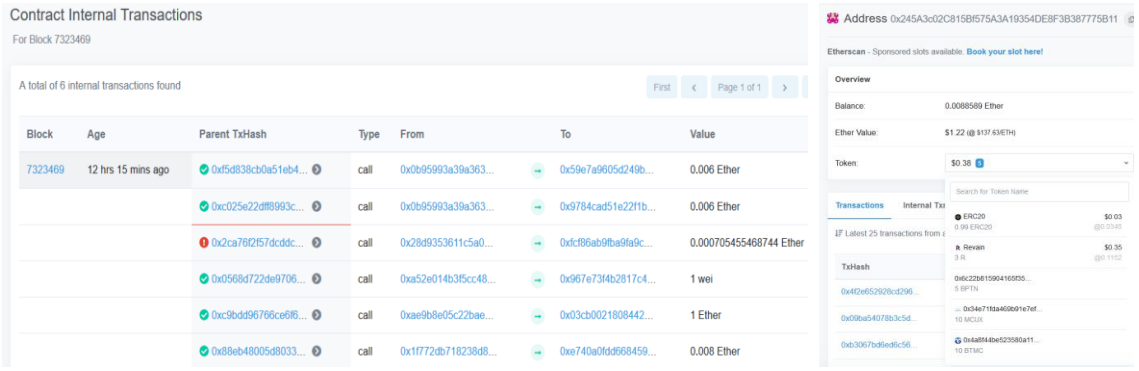
Smart-contract is a programming code written in a special programming language - Solidity. It is run by the Ethereum Virtual Machine (EVM) [9]. Similar to the standard contracts that are concluded each day, in the smart contract can be described clauses to be fulfilled upon the occurrence of a certain event. But, in the case of smart-contracts, the logic described in the clauses can be executed automatically. No third party is required to monitor compliance with the terms of the contract. Considered to be a drawback, but also for the biggest advantage of decentralized smart-contracts, is

the impossibility of changing the contract code after it has already been broadcast on the network. The only one possible option after uploading of the existing smart-contract is uploading of a new version, but it has to tell everyone who uses old version that they have to turn to the new one.

Nowadays, smart-traders are finding a wider choice in buying and selling with smart-contracts. When the agreed amount, which is described in the Smart Contract, arrives at its address, ownership is transferred to the buyer's public key, and the amount is automatically sent to the seller. In these days the broadest use of smart-contracts are found in ICO campaigns. In this type of campaign, investors, translate Ethers to the contract address and in return they receive tokens. In this way a Start-up project can raise capital to start its business. In most cases, these tokens give special rights to their owners within the project they invested in. They can also be integrated by project creators as an exchange unit in the project, or to act as shares in a company.

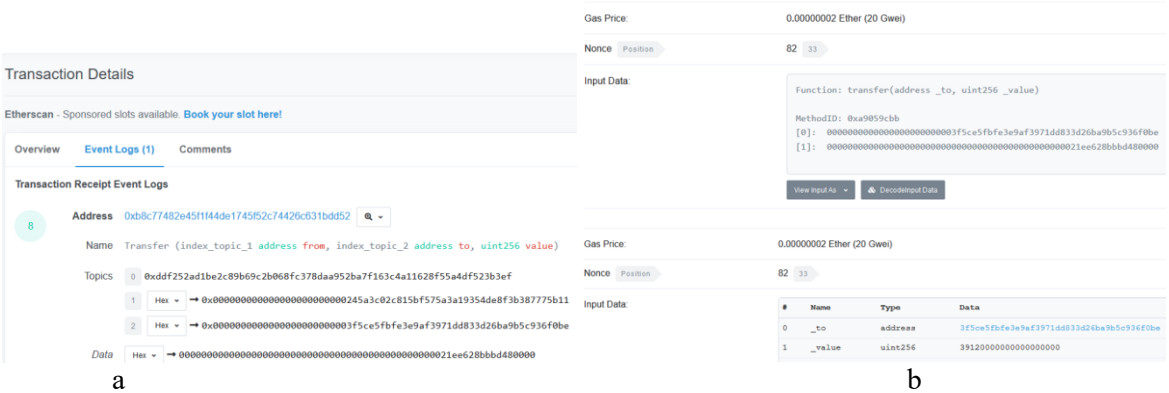
The tokens can be stored in Ethereum wallets, and they must meet the ERC20 standard [11]. Despite of wide support of this standard, there are also wallets that do not support it. Ethereum is a suitable Token Maker system because it is one of the most widely used block-chain currencies. There are well-established token implementation standards and the most widely supported Block-chain system on the stock exchanges where it can deposit, withdraw and trade tokens.While in Bitcoin block-chain diggers only validate bitcoins, in the Ethereum block-chain using the Ethereum virtual machine, they perform smart contract functions and can reach the available amount of tokens at each address and validate or deny the token transfer.

One example of transaction of crypto- tokens in details is presented on the Fig. 2 and Fig.3. On the Fig.2a it is presented relationship between the client "From:" and the smart contract "To:". The client pays transaction fee to process this transaction from the diggers and get it into a block. In Event Log (Fig.3a) it is visible which function of the smart-contract is called to effect the transfer of crypto-tokens as well as its parameters. In overview (Fig.2b) it is visible the parameters of the Ethereum transaction itself.



a b

Figure 2.Block, where transaction of tokens (a) and details (b) is made.



a b

Figure 3.Event Logs of the crypto-token transaction (a), and GAS price for this transaction (b).

2.3. Phases in the Development of Smart contract

The development of a smart contract takes place in 3 phases:

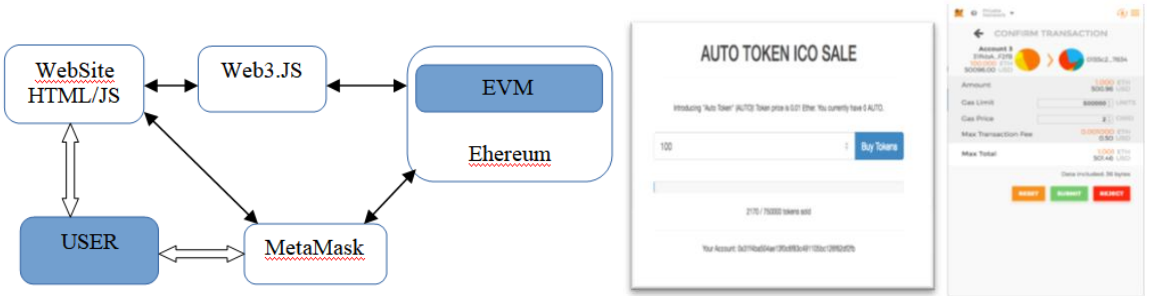
1. Analyze - Which problem needs to solve a smart contract. The main objects, as well as the links between them, are identified.

2. Design - Based on the information from the first phase must develop schema of the objects that will serve as variables and the interactions with other objects that will be realized as functions. The information gathered also serves to determine the block-chain technology at which a smart contract will be realized.

3. Implementation - The components of the smart-contract are implemented and broadcast on the network.

3. Implementation of proposed Smart contract

Architecture of application is presented on Fig.4a. The implementation of the ERC20 standard guarantees the smart-contract free movement between the participants in the Ethereum network. The proposed solution was developed with Truffle and Ganache under the MacOS High Sierra operating system. Ganache creates a local block-chain based on Ethereum, which can directly execute commands as well as perform tests.



a b

Figure 4.Architecture (a) and interface (b) of application.

Metamask is used, because there is no need to download a local copy of the entire block-chain, and when it is making a connection to a site that makes a connection with Ethereum. Metamask takes care of all requests from and to the block-chain network. Metamask can perform the Ethereum wallet function and support sending and receiving Ethers and ERC20 tokens. The private key of the users is stored on the local machine, so Metamask's servers have no information about it. An interface view is shown on Fig.4b. The application interface is based on JavaScript library web3.js, which allows communication with the Ethereum network and perform smart contracts.

Truffle is used for the implementation of the smart contract. It is an integrated system for compilation of the written smart contracts, and its uploads on the Ethereum network. In addition to the functions implementing the ERC20 standard, to ensure the proper operation of the smart-contract, the functions for token sale, termination of sale and payment of funds to the contract owner are needed (Fig.5).

```
function multiply(uint x, uint y) internal pure returns (uint z) {
    require(y == 0 || (z = x * y) / y == x);
}

function buyTokens(uint256 _numberOfTokens) public payable {
    require(msg.value == multiply(_numberOfTokens, tokenPrice));
    require(tokenContract.balanceOf(this) >= _numberOfTokens);
    require(tokenContract.transfer(msg.sender, _numberOfTokens));

    tokensSold += _numberOfTokens;

    Sell(msg.sender, _numberOfTokens);
}
```

a

```
function endSale() public {
    require(msg.sender == admin);
    require(tokenContract.transfer(admin, tokenContract.balanceOf(this)));

    admin.transfer(address(this).balance);
}
```

b

Figure 5. Purchase of tokens (a), Termination of sale (b) .

When calling the function „endSale“, it is checked who call it, because only the account administrator has the rights over it. After that, all remaining unsold tokens are transferred to the account of the administrator' on its address. As a final step, all collected Ethers are transferred to the administrator. This way, if someone tries to make a purchase after calling "endSale," check for available tokens will fail and will not be sold.

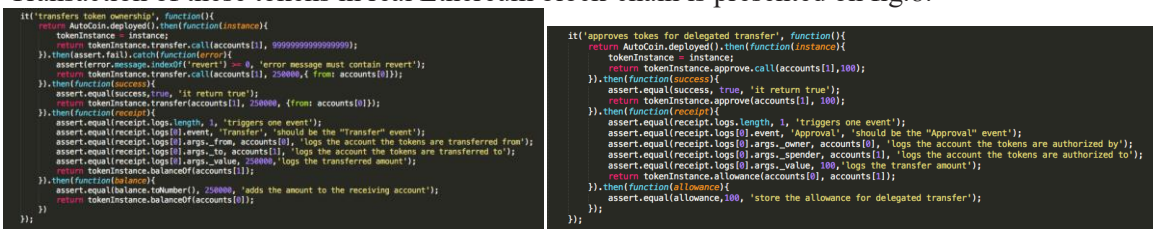
4. Experimental results

First part of the tests is related to the proper work of smart contracts- balance of account, transfer of tokens etc. There are a handful of tools for automated smart contract (written in Solidity) security vulnerability testing based on code-level analysis. In [8] is given a synopsis of the four most related tools that is possible to use in experiments, namely Oyente, Mythril, Securify, and SmartCheck. However, the evaluate level of rigor, ranging from syntactic, heuristic, analytic to fully formal, refers to underlying security testing technique of the given tool and up to this moment the researchers trusted on the implemented in the Solidity test tools. Truffle (and Solidity) has a built-in smart-contract-testing mechanism that is written in JavaScript, which here is used.

On Fig. 6a a token transfer test code is provided. For direct transfer testing, 250 000 tokens are transferred from the administrator's address to the recipient's address. Once the transfer has taken place, the event is captured and checked for "Transfer" type. If this test is successful, the balance of the recipient address is checked for the presence of transferred tokens.

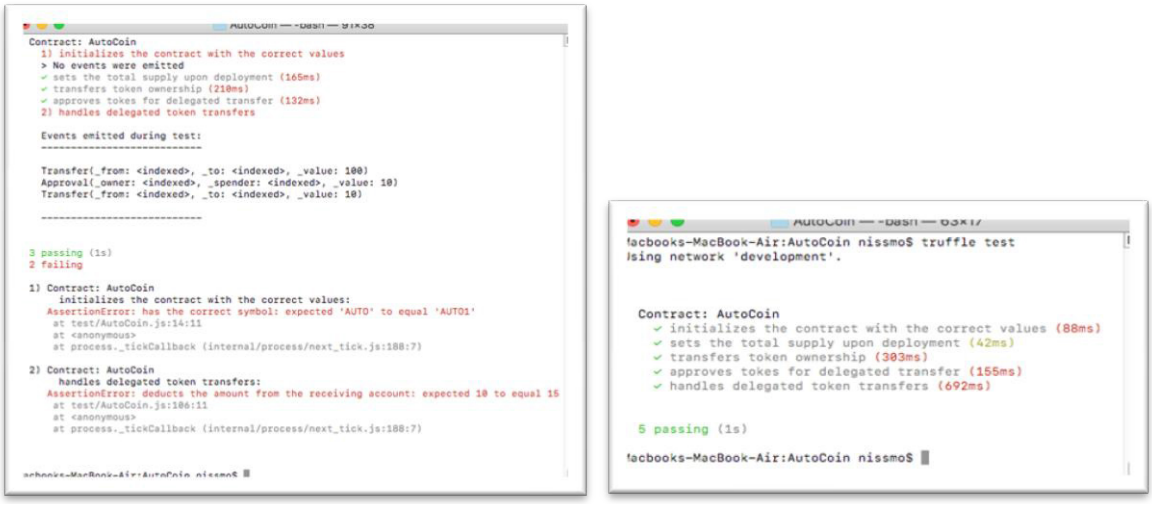
On Fig. 6b a test code for crypto-token disposal is provided. The delegated transfer check is similar to the direct transfer check. First, 100 tokens are transferred from the administrator's address to the address from which a delegated transfer will be allowed - address 1. It is allowed 10 tokens to be spent from address 3, which sends them to address 4. After performing these actions, it is expected that address 1 to have 90 tokens, address 2 - 0, and address 3 to be 10 tokens. The result of their implementation with wrong parameters is shown in Fig.7a, and with the correct parameters - in Fig.7b.

Transaction of these tokens in real Ethereum block-chain is presented on fig.8.



a b

Figure 6. Code of test for transfer of tokens (a), Code of test for disposing of tokens (b) .



a b

Figure 7. Results of test with wrong parameters (a), proper parameters (b).

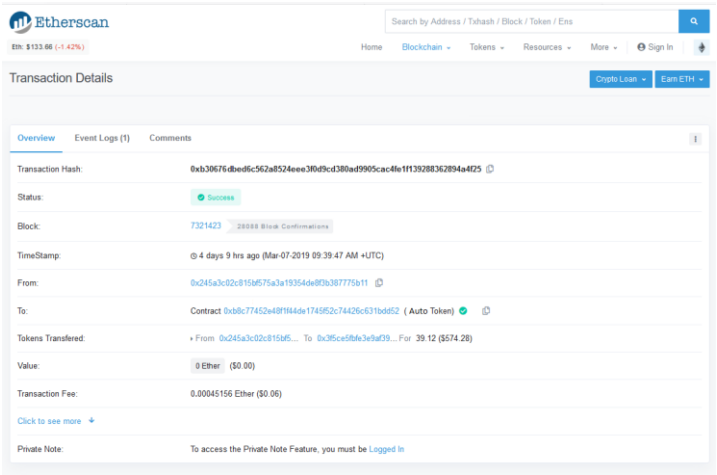


Figure 8. Transaction of the proposed token in the Ethereum block-chain.

5. Network communication related to proposed token for Smart contract

In the real Ethereum block, as much as power to include block time is fixed, there comes a dynamic change of difficulty depending on how much power is included in the network. The tests were carried out in the local network with flat topology. The client connects to the Metamask server. The parameters of computers are Apple Mac Book Pro Late 2011 Specs, Core i5 (I5-2435M) 2.4GHz 2/4 Cores/Threads, 4GB DDR3 1333Mhz RAM. In the Metamask when sending an ether to buy a token, there is used protocol TLSv1.2.

In Fig.9 is presented network communication between client and Metamask server during successful transaction of tokens. The client send Ethereum to the Metamask server to buy tokens. The client PC has IP address 192.168.100.2 in the local network in Varna, Bulgaria. Data transfer between the client and Metamask server is encrypted. The server is located in Ashburn, United States with an IP address 34.202.251.153 and communication is over TLSv1.2 protocol. The Metamask server does not allow invalid transactions. In this case, the block time is set so that when a transaction is created, a new block will be created and entered immediately into.

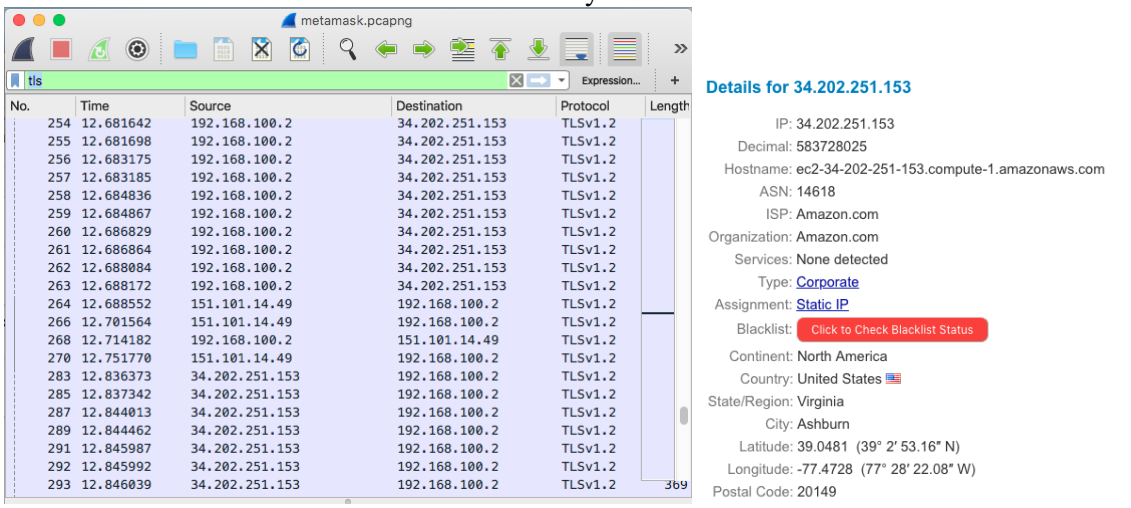


Figure 9. Network communication from client to Metatask during transaction of the proposed token

6. Conclusions

In this paper is proposed a web-based solution for the creation of a decentralized token for the implementation of a smart contract based on Ethereum block-chain. The implementation of the ERC20

standard guarantees its free movement between the participants in the Ethereum network. There are presented experimental tests, which prove proposed crypto-token validity.

References

[1] Top 100 Crypto currencies by Market Capitalization, https://coinmarketcap.com/ (last visit on 08.04.2019)
[2] Bitcoin is already the stock exchange on Wall Street, https://hicomm.bg/kratki-novini/bitkoin-veche-e-na-borsata-na-uol-striit.html (last visit on 08.04.2019)
[3] Cboe Global Markets Inc., http://www.cboe.com/ (last visit on 08.04.2019)
[4] CME Group Inc., https://www.cmegroup.com/ (last visit on 08.04.2019)
[5] Olga Labazova, Tobias Dehling, From Hype to Reality: A Taxonomy of Blockchain Applications, Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019), DOI: 10.24251/HICSS.2019.552
[6] Balázs Bodó; Daniel Gervais; João Pedro Quintais, Blockchain and smart contracts: the missing link in copyright licensing?, International Journal of Law and Information Technology, Volume 26, Issue 4, Winter 2018, Pages 311–336, https://doi.org/10.1093/ijlit/eay014
[7] Florian Idelberger,Guido Governatori, Regis Riveret,Giovanni Sartor, Evaluation of Logic-Based Smart Contracts for Blockchain Systems, RuleML 2016, DOI: 10.1007/978-3-319-42019-6_11
[8] Reza M., PariziAli Dehghantanha, Kim-Kwang Raymond Choo, Amritraj Singh, Empirical Vulnerability Analysis of Automated Smart Contracts Security Testing on Blockchains,2018, Proceedings of CASCON'18, October 2018, Canada.
[9] Ethereum Homestead Documentation, http://www.ethdocs.org/en/latest/ (last visit on 08.04.2019)
[10] Ethereum Blockchain Explorer, https://etherscan.io/ (last visit on 08.04.2019)
[11] ERC20 Tokens list, https://eidoo.io/erc20-tokens-list// (last visit on 08.04.2019)