Towards a Blockchain-based Software Engineering Education

Filipe Fernandes[1,2], **Cláudia Werner**[1]

[1]Universidade Federal do Rio de Janeiro – COPPE/UFRJ
Systems Engineering and Computer Science Department
Caixa Postal 68.511 – 21.941-972 – Rio de Janeiro – RJ – Brazil

[2]Federal Institute of Southeast Minas Gerais – IF Sudeste MG
Computer Science Department
Caixa Postal 413 – 36.909-300 – Manhuaçu – MG – Brazil

{ffernandes,werner}@cos.ufrj.br

Abstract. *Blockchain technologies for rewards in education are gaining attraction as a promising approach to motivate student learning and promote academic achievement. By providing tangible rewards for educational attainment and engagement, such as digital tokens, educators can motivate learners to take a more active role in their learning and increase their sense of ownership and responsibility for their academic outcomes. In this context, this work proposes the Software Engineering Skill (SES) token as a way of rewarding students in order to improve their experiences in Software Engineering Education (SEE). We performed a proof of concept and conclude that SES token can be deployed in a platform to support SEE.*

1. Introduction

Blockchain technology in education can provide equal opportunities for students to develop themselves through the implementation of a token economy in different classroom settings [Tan et al. 2022]. The adoption of a token economy is perceived as an essential mechanism for sustaining interest and engagement in the educational process. For this reason, this work proposes the Software Engineering Skill (SES) token as an approach to support blockchain-based Software Engineering Education (SEE). SES token aims to engage and motivate students and developers in order to improve experiences in SEE, as well as increase developer contributions on the educational platform providing SES tokens.

This paper is organized as follows: Section 2 presents the research context of which this work is part. Related works are described in Section 3. Our proposal is described in Section 4 and a proof of concept is presented in Section 5. Finally, our conclusions and future directions are described in Section 6.

2. Research Context

This work is an extension of a doctoral thesis in progress that proposes an approach to enable the Metaverse-based Software Engineering Education (MetaSEE) [Fernandes and Werner 2023]. MetaSEE approach aims to provide an interoperable and scalable structure composed of the Metaverse's main concepts and technologies grouped in five layers: Physical, Virtual, Metaverse Engine, MetaSEE, and Infrastructure. Figure 1 presents a structure overview to enable the MetaSEE.

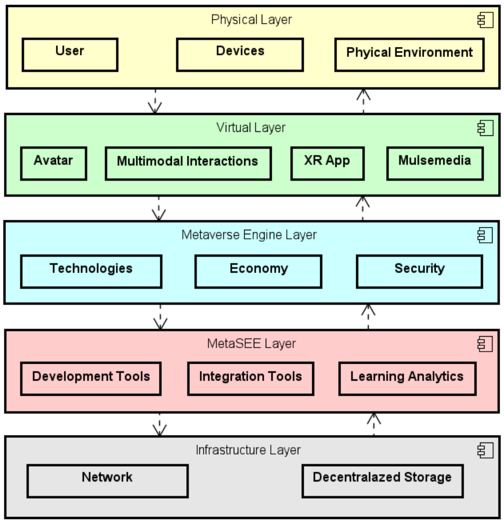


Figure 1. MetaSEE structure overview

Physical Layer corresponds to the main entities external to the Metaverse that belong to the physical and real world. *Virtual Layer* establishes the main components of the "virtualization" of physical layer elements. *Metaverse Engine Layer* is composed of general *Technologies*, as well as *Economics* and *Security* of the Metaverse. *MetaSEE Layer* is the main contribution to support SEE through the Metaverse. *Development Tools* component should provide a set of mechanisms to facilitate the development of XR applications (XR apps - known as immersive applications) for SEE considering the range of complexity and characteristics involved. *Integration Tools* component should provide reusable functionality for SEE. *Learning Analytics (LA)* is the component that must guarantee the maintenance of the learning performance of the Metaverse for SEE users. *Infrastructure Layer* deals with network and decentralization aspects of the Metaverse. In order to contribute to the MetaSEE approach, more specifically in the *Decentralized Storage* component of *Infrastructure Layer*, this work proposes to use blockchain technology to create a tokenization mechanism, which is detailed in Section 4.

3. Related Work

Multiple institutions of higher education have integrated blockchain technology into their systems to devise various solutions and strategies pertaining to higher education, more precisely in certificate authentication and academic verification [Turkanović et al. 2018]. In addition, blockchain technology can be applied as an awarding system for performance assessment in education using blockchain technology. [Alammary et al. 2019] present a system in which learners will be awarded badges, which are certified for a predetermined level of progress in terms of learning. Inspired by previous work, our proposal is to fill a significant gap in the literature, since no previous work was identified with the purpose of supporting SEE through blockchain technology.

4. Blockchain-based Software Engineering Education

The main goal of this work is to propose a solution that meets the *Decentralized Storage* component in order to contribute to the MetaSEE approach. This work designs a mechanism through tokenization with the purpose of fostering a digital reward in order to improve student engagement, as well as motivating developers to contribute extensions (plugins) to the MetaSEE approach.

Therefore, we defined the SES as a MetaSEE token to engage and motivate students and developers. Considering the MetaSEE approach context, SES token can provide a mechanism based on "learn to earn" for students [Tan et al. 2022].

The use of SES token can provide tangible incentives for students, which can be used as rewards for achieving learning goals or performing other activities within the platform. In addition, creating a reward system can help making the learning process more playful and engaging, encouraging students to dedicate themselves more to their studies.

SES token can also help make the platform more interactive and collaborative. For example, students can be rewarded for helping other students solve problems or answer questions. This can create a more engaged and collaborative community of students, which can be an additional source of motivation for learning. Additionally, SES token can be used as a way to record and recognize student achievements. For example, successful completion of an activity can result in a token being issued, which can be used to unlock additional content or access special features. This can help create a sense of accomplishment and progress for students, which can motivate them to keep learning and progressing.

Another way and use of the SES token is in the gamification of the learning experience, making the MetaSEE platform more attractive and engaging. By setting clear goals and rewards for students, SES token can help create a more exciting and challenging learning environment that can keep students motivated and engaged longer.

For developers, SES token can be an effective mechanism to motivate developers to contribute to the MetaSEE platform. Developers can receive tokens as a reward for contributing new extensions and bug fixes. These rewards can provide a form of recognition for work done and can be used to encourage a culture of collaboration in the developer community. This can result in increased participation as well as increased Software Engineering (SE)-specific features through extensions.

5. Proof of Concept

In this section, we present the SES token implementation and consider this step as Proof of Concept (PoC). Our implementation was developed with JavaScript (JS) and Solidity[1] languages, as well as Hardhat[2] framework. Solidity is a smart contract programming language used on the Ethereum platform and Hardhat is a software development environment for Ethereum that allows testing, compiling, and deploying smart contracts. Hardhat is a useful tool for developers working with Solidity as it simplifies and automates many common smart contract development tasks.

In order to prove a SES token basic transaction between accounts, we implemented a smart contract written in Solidity that implements a custom ERC20 token called *SESkillToken*, as shown in Figure 2. ERC20 defines a set of interfaces and standards for implementing digital tokens on the blockchain. This means that *SESkillToken*, being a

[1]https://soliditylang.org/
[2]https://hardhat.org/

contract that implements the ERC20 standard, follows ERC20 interfaces and standards. This ensures that *SESkillToken* is compatible with other tokens and applications that also implement the ERC20 standard. Lines 5 to 7 import contract inherits functionality from other OpenZeppelin[3] (open-source repository with standard contracts) contracts such as *ERC20*, *ERC20Capped* and *ERC20Burnable*, which means that the *SESkillToken* has the same functionality as those contracts, in addition to the custom functionality defined in this contract.

```
1   // SPDX-License-Identifier: GPL-3.0
2
3   pragma solidity ^0.8.18;
4
5   import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
6   import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Capped.sol";
7   import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol";
8
9   contract SESkillToken is ERC20Capped {
10    address payable public owner;
11    uint256 public blockReward;
12
13    constructor(uint256 cap, uint256 reward) ERC20("SESkillToken", "SES") ERC20Capped(cap * (10 ** decimals())) {
14      owner = payable(msg.sender);
15      _mint(owner, 70000000 * (10 ** decimals()));
16      blockReward = reward * (10 ** decimals());
17    }
18
19    function _mintMinerReward() internal {
20      _mint(block.coinbase, blockReward);
21    }
22
23    function _beforeTokenTransfer(address from, address to, uint256 value) internal virtual override {
24      if(from != address(0) && to != block.coinbase && block.coinbase != address(0)) {
25        _mintMinerReward();
26      }
27      super._beforeTokenTransfer(from, to, value);
28    }
29
30    function setBlockReward(uint256 reward) public onlyOwner {
31      blockReward = reward * (10 ** decimals());
32    }
33
34    function destroy() public onlyOwner {
35      selfdestruct(owner);
36    }
37
38    modifier onlyOwner {
39      require(msg.sender == owner, "Only the owner can call this function");
40      _;
41    }
42  }
```

Figure 2. SES token code

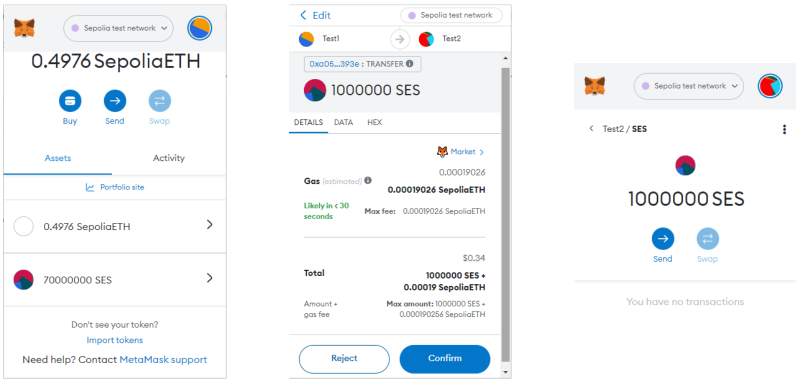
Line 13 defines a constructor that establishes the token's name and symbol, as well as its bid limit. In addition, it defines the contract owner's address and the miner reward amount per block. The *_mintMinerReward()* function (line 19) is responsible for issuing new coins to the miner who mined the current block, with a value equals to the one defined in the *blockReward* variable. On line 23, *_beforeTokenTransfer()* function is a hook that runs before each token transfer. In this case, it checks if the transfer is taking place between an address different from the address of the block where the transfer is taking place. If this is true, then it calls the *_mintMinerReward()* function to issue the miner reward.

The contract also has other functions such as *setBlockReward()* which allows the contract owner to set the miner reward amount per block and the *destroy()* function which allows the owner to destroy the contract and recover the remaining funds. Finally, the contract uses an *onlyOwner* modifier to ensure that only the owner of the contract can call functions that require owner privileges, such as *setBlockReward()* and *destroy()*.

[3]https://www.openzeppelin.com/

Smart contracts are programs that will run on the blockchain, which means that once deployed, they will be immutable. Any error or problem in the contract can be difficult or even impossible to fix, so it is important to ensure that the contract is working correctly before deploying it to the network [Zheng et al. 2017]. For this reason, we implemented unit tests to verify that the SES token functions behave as expected and that the variables are updated according to the contract specifications.

After all the unit tests passed successfully, we performed a basic integration test with the MetaMask[4] wallet. MetaMask is a browser extension that allows users to access Decentralized Applications (dApps) based on blockchain, such as Ethereum, directly from their web browsers. It is a digital wallet that enables users to store, send, and receive cryptocurrencies, as well as interact with decentralized applications [Zheng et al. 2017]. Figure 3 presents the transfers between test accounts.



(a) Account 1 balance (b) Transfer confirmation (C) Tokens received on account 2

Figure 3. SES token transfer between test accounts on MetaMask wallet

Firstly, we added the Sepolia[5] testnet in the wallet and integrated it with the SES token. A testnet is a network in the blockchain ecosystem that allows developers and users to test and experiment with blockchain applications and smart contracts in a simulated environment that closely resembles the real blockchain network. From the SES local token project on the desktop, 70,000,000 tokens were sent to test account 1, as shown in Figure 3 (a). According to Figure 3 (b), there is a transfer confirmation between accounts of 1,000,000 tokens. After the confirmation, these tokens are transferred to test account 2, as shown in Figure 3 (c).

In order to prove these operations, these transactions were searched on Etherscan[6]. It is a popular blockchain explorer for the Ethereum network that allows users to view and search for transactions, addresses, and other activities on the blockchain. According to Figure 4, transfer transactions were performed successfully.

6. Conclusion

This work described a token to support blockchain-based SEE, named SES. More precisely, we performed a PoC in order to verify the feasibility from a technological point of view, and the entire implementation was tested on the Sepolia testnet.

[4]https://metamask.io/
[5]https://sepolia.dev/
[6]https://etherscan.io/

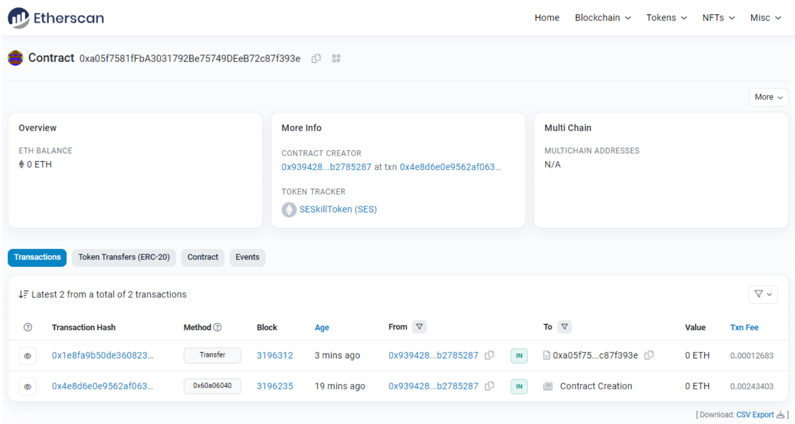


Figure 4. Transactions report on the Etherscan

SES token-based rewards in education presents a range of advantages for learners and educators. SES token provides a tangible incentive for learners to engage in educational activities and achieve learning outcomes and can create a more engaged and collaborative community of students, which can be an additional source of motivation for learning.

Although the implementation of token-based rewards in education presents several advantages, it also entails several disadvantages that should be considered. One of the primary concerns is that students may become more focused on earning tokens rather than on the learning process itself, which can lead to a decrease in the quality and depth of learning. Another concern is the potential for token-based rewards to reinforce existing power structures and inequalities within the education system. For instance, learners who have greater access to resources or who are already performing well may be more likely to earn tokens, while those who are disadvantaged or struggling may be left behind.

As future works, we intend to perform research to solve token-based rewards concerns, the evolution of the SES token implementation in order to integrate with the MetaSEE platform, as well as evaluation with students, educators, and developers.

References

Alammary, A., Alhazmi, S., Almasri, M., and Gillani, S. (2019). Blockchain-based applications in education: A systematic review. *Applied Sciences*, 9(12).

Fernandes, F. A. and Werner, C. M. L. (2023). A Scoping Review of the Metaverse for Software Engineering Education: Overview, Challenges and Opportunities. *PRESENCE: Virtual and Augmented Reality*, pages 1–40.

Tan, K. H., Kasiveloo, M., and Abdullah, I. H. (2022). Token economy for sustainable education in the future: A scoping review. *Sustainability*, 14(2).

Turkanović, M., Hölbl, M., Košič, K., Heričko, M., and Kamišalić, A. (2018). Eductx: A blockchain-based higher education credit platform. *IEEE Access*, 6:5112–5127.

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564.