DeLMS: A decentralized learning management system using Ethereum smart contracts and IPFS

Midhun Xavier *, Parvathy Sobha †, Sandeep Patil*, Valeriy Vyatkin* ‡
* Department of Computer Science, Computer and Space Engineering, Lulea Tekniska Universitet, Sweden
† Department of Engineering Sciences and Mathematics, Lulea Tekniska Universitet, Sweden
‡Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland

Email: midhun.xavier@ltu.se, parvathy.sobha@ltu.se, sandeep.patil@ltu.se, vyatkin@ieee.org

2023 IEEE 21st International Conference on Industrial Informatics (INDIN) | 978-1-6654-9313-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/INDIN51400.2023.10217957

***Abstract*—A learning management system (LMS) is a web-based technology developed to improve the learning process in educational institutions. This study addresses the drawbacks of existing LMS by developing DeLMS (Decentralized Learning Management System) and aims to revolutionize the way educational institutions manage and deliver educational content using Ethereum smart contracts and the IPFS (InterPlanetary File System). By leveraging Smart contracts and IPFS cutting-edge technologies, DeLMS provides a secure, transparent and decentralized platform for managing and delivering educational content. The content addressing feature of IPFS eliminates the need for constant requests to centralized servers for the location of files, resulting in faster file transfer and reduced energy consumption. The proposed system has the potential to change the way educational institutions operate, making it more accessible and beneficial for both students and educators.**

***Index Terms*—Blockchain, Ethereum Smart Contracts, IPFS, NFT**

I. Introduction

LMS is defined as a web-based technology developed to improve the learning process through its proper planning, application, and evaluation in educational institutions [1]. The very first idea of LMS was first developed by Sidney Pressey in 1924 when he invented the 'teaching machine' [2]. LMS in its present shape was introduced around the late 1990s. Currently, LMS systems are widely used across the globe and constitute a larges share of the learning system market [3]. Most importantly, LMS has faced extensive growth due to the shift to remote learning during the COVID-19 pandemic [4].

The authors accept that data sharing or data exchange between systems, databases, and applications is crucial in the present world in all fields including the education system. Data interoperability provides the opportunity for equipment, systems, apps, or products from different suppliers, regions, or institutions to operate together in an organized way, by facilitating unrestricted sharing of data, between systems via local area networks (LANs) or wide area networks (WANs). However, this ability to share effortlessly from one user to another comes with a cost.

The traditional LMS typically store data on centralized servers, this makes the system insecure, unsustainable and vulnerable. It is possible that hackers can gain access to the centralized server, they could change the content of the website or they could potentially access sensitive information such as student's personal information and grades. Another major problem that confronts traditional LMS is their sustainability, as centralized systems require significant energy consumption to run and maintain, which can result in a significant carbon footprint. Furthermore, because the data is stored in a single location, LMS may require more energy for data transfer. In contrast, a decentralized approach to information management can help reduce energy consumption by limiting the distance the information has to travel.

In this study, the authors try to address these problems by developing a DeLMS to secure and decentralised ways to manage educational content and records, as well as potentially reducing costs associated with traditional centralized systems using Smart contracts and IPFS. This research focuses on developing a decentralized and secure repository of a learning management system. The developed framework can be extended to include broader applications in the education field requiring low to high data security. Examples of applications include data sharing by students (submission of assignments, theses, articles, sharing files, sending messages, writing comments etc.,), which requires low to medium security, and data sharing by researchers and project facilitators (project proposals, project quotation, project plan, funding proposals, etc.,) which requires medium to high-security data sharing.

The paper is organized as follows: Section II discusses the tools and technologies used and background. Section III explains the methodology to design and develop a decentralized LMS application using IPFS in detail. Section IV describes the illustrative example of a content submission platform using an Ethereum smart contract and presents the result of the work. Finally, Section V concludes the paper and outlines future goals.

II. Background

The proposed solution includes multiple key aspects like blockchain, smart contract, IPFS protocol, etc. This section gives a brief description of these concepts, and other tools and technologies used to develop the learning management system.

A. Blockchain

The concept of blockchain was first implemented in peer-to-peer electronic cash systems to improve reliability by time stamping transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work [5]. Blockchain has much wider applications and is defined as a decentralized, distributed, and immutable ledger to store digital transactions [6]. Blockchain constitutes of an expanding list of records named blocks, which are linked and stored using cryptography [6]. The major advantage of blockchain is the decentralized framework promoting secure and reliable operations [7]. However, existing blockchain networks possess a disadvantage. It does not allow to build and deploy decentralized applications on its network, which again reduces the system's reliability.

B. Ethereum

Ethereum technology overcomes the disadvantage of the existing blockchain networks. Ethereum is a blockchain based technology for building decentralized applications using smart contracts. Ethereum network operates without being controlled by a central authority. As Ethereum is programmable, decentralized applications can be built and deployed on its network [8].

1) Smart Contract: A smart contract is a script that is anchored on a blockchain or similar distributed infrastructure. When triggered by a blockchain transaction and validated across the network, predefined actions are executed [9]. Since the conditions of a smart contract are transparently stored on the blockchain, it will always operate as all parties intend, which can reduce trust issues between the involved parties. Smart contracts are software scripts, just like scripts that run on non-blockchain applications [10]. Ethereum Virtual Machine (EVM) is employed to run the smart contract.

C. InterPlanetary File System

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files [11]. IPFS provides a high throughput content-addressed block storage model, with content-addressed hyperlinks. This forms a generalized data structure upon which versioned file systems, blockchains, etc can be built. IPFS combines a distributed hashtable, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other [12].

The literature includes the development of various systems that employ a wide variety of applications using blockchain, particularly for data storage. Researchers have developed systems that can store data from IoT services where the transactions are managed by the blockchain through smart contact [13]. Blockchain-enabled systems have been proposed to verify the credibility of online news using ethereum smart contracts and IPFS [14]. Researchers have developed a system to manage online user reviews using ethereum block chain and IPFS applications [15]. Different data sharing methods using block chain applications have been proposed in the literature including mechanisms for sharing power material supply chain data [16], ocean data [17], equipment data [18], etc. Literature includes data storing and sharing applications for the educational field as well. Researchers have proposed an ethereum blockchain and IFPS based system for authenticating educational documents [19], block chain based system for tracking and fraud detection of educational certificates [20]. However, to the best knowledge of the authors, no work has been conducted on developing ethereum block chain and IFPS based system with a focus on learning management systems. The proposed work developed a solution to address the existing problem of LMS to transform the existing LMS into a more reliable, secure, and sustainable system.

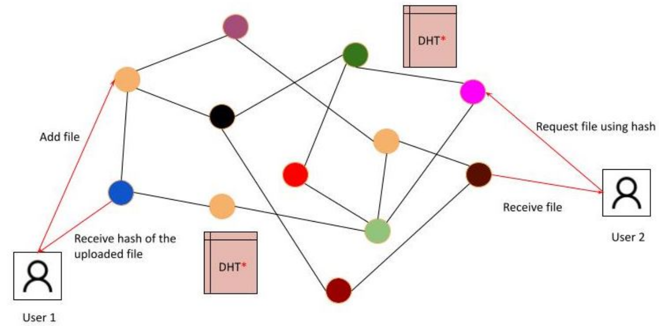

Fig. 1. IPFS network cluster

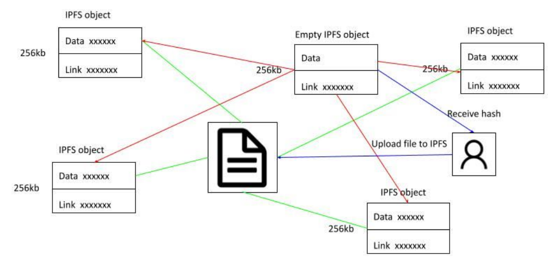

Fig. 2. IPFS data store

III. Methodology

The research addresses drawbacks in existing LMS by developing a blockchain based content sharing system using IPFS. It involves the development of a decentralized secure repository for the LMS. The following subsections describe how key aspects, the blockchain, and IPFS protocol are incorporated in LMS.

A. Using IPFS protocol for data handling in LMS

The IPFS protocol uses content-addressing to uniquely identify each file in the system. "What to find?" is answered using the protocol rather than "where to find?" for which each file uploaded in the system is given a unique hash id called content Identifier (CID) which is considered as a fingerprint. The hash id changes when the data is changed [21]. Figure 1. describes the IPFS network cluster. The below steps show the

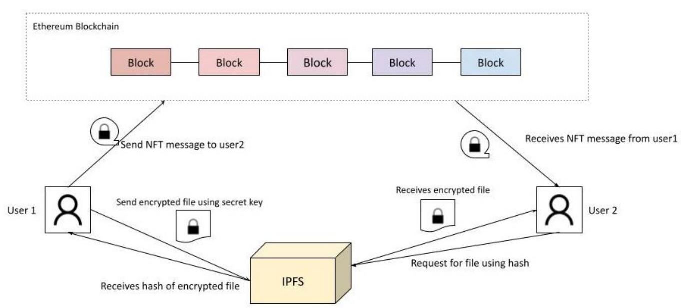

Fig. 3. Framework of Block-chain enabled IPFS data handling system

chronological sequence of actions for storing and retrieving data in an IPFS network cluster. The LMS user can be a student or teacher registered in the LMS.

Step 1: User 1 uploads the file into the IPFS enabled system.
Step 2: The data is encrypted using a secret key and stored in IPFS system (Figure 2. explains the data storing in IPFS network in detail)
Step 3: A hash id is generated for the data and is received by user 1.
Step 4: In order to retrieve the data, user 2 receives the hash id along with the secret key to decrypt the file from user 1. This information is encrypted using user 2's public key.
Step 5: User 2 decrypts the information using user's private key and obtains the hash id and secret key to decrypt the file.
Step 6: User 2 requests IPFS for the file using the received hash id, receives the file, and decrypts the data.
Step 7: Once decrypted the data, user 2 can generate a new hash id and compare it with the original hash id received from user 1.

The data uploaded by the user is stored in IPFS objects. Figure 2. describes the data storage using the IPFS technique. Each IPFS object can store 256 kb. If the data is more than 256 kb, the data is divided into IPFS objects and linked to other empty IPFS objects. Each IPFS object has two parts, one for storing data and the other for storing the next IPFS object link where the following data link is stored.

1) Implementing block chain enabled solution in IPFS data handling system: Figure 3. shows the framework of the Blockchain enabled IPFS data handling system. In this framework, blockchain technology is employed as the communication ledger. The generated hash id (when user 1 uploads a file into the system) along with the secret key is used to encrypt the data and is sent to user 2 as a non-fungible token (NFT) message through the blockchain network.

B. Architecture and Work flow

Figure 4. shows the integrated framework of the proposed architecture including blockchain technology and IPFS protocol. The below steps describe the sequence of actions while using the framework.

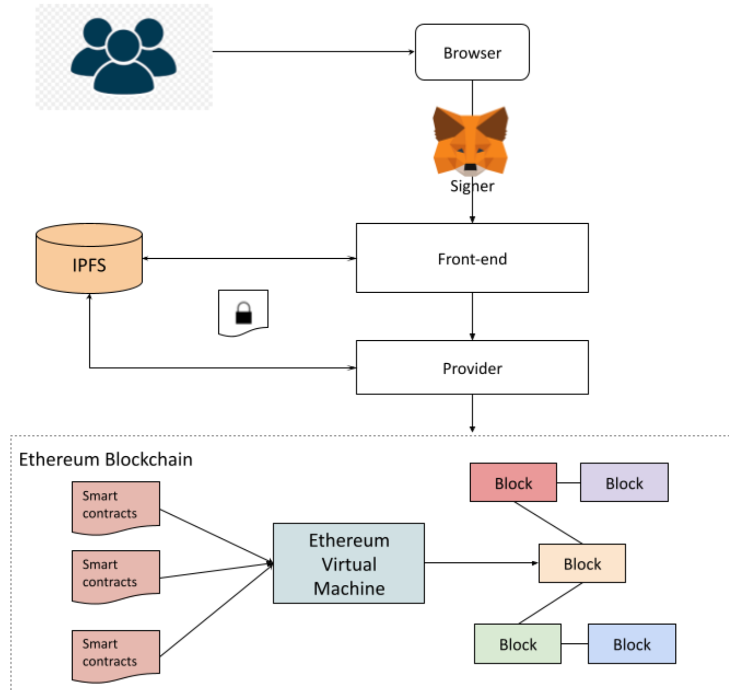

Fig. 4. Architecture of blockchain enabled IPFS data handling system

Step 1: User 1 reaches the user interface (UI) via browser.
Step 2: User 1 uploads the file to the system.
Step 3: User 1 can view the uploaded files in UI.
Step 4: User 1 receives the unique hash id.
Step 5: The hash id and secret key to decrypt the file is sent as NFT message to the other user 2 (to whom the file has to be shared) via block chain.
Step 6: User 2 receives a notification in the user interface.
Step 7: User 2 receives the NFT message and requests the file in the IPFS system.
Step 8: User 2 receives the file and user 2 can view the shared files in UI.

Note
Actions in user interface
Actions in backend

If no action is to be performed by user 1 after uploading the file, user 1 can exit the portal. If a new user, user 3 needs the file, user 3 can request user 1 and user 1 can accept or reject the request.

C. Addressing the problem

DeLMS is a decentralized application, and hence lacks a centralized point of control. This increases the reliability of the blockchain-enabled LMS system, reducing failure rates. Also, decentralization increases the transparency of the system. This feature makes the LMS more robust to failures. IPFS enables generations of hash ids for the files uploaded to LMS. This hash id changes when the data is changed hence the data cannot be tampered with. This feature would be highly relevant while storing data with medium to high security. Also, this feature adds more security and makes the system immutable. When the same file is duplicated in LMS, the hash id generated for the newly uploaded file will be the same as that of the previous file, and hence no content duplication is possible. This feature adds to efficient data storage and is significant from a sustainability point of view. DeLMS that uses IPFS for file storage can reduce the energy consumption associated with data transfer. IPFS allows data to be stored on multiple nodes, so the data can be accessed from a node that is closest to the user, reducing the amount of energy needed for data transfer. In addition, DeLMS uses blockchain technology to reduce the need for constant maintenance and monitoring of central servers, further reducing energy consumption.

IV. Case study

Content submission and evaluation of records, assignments, theses or any kind of data to be evaluated or shared with anyone is one of the major use cases of LMS and it is taken as an illustrative example to evaluate the possibilities of the proposed solution. In this paper, a decentralized application for managing courses, sending messages, posting articles, submitting, evaluating and sharing educational content is developed. Smart contracts are reusable snippets of code and users make requests to execute these code snippets with varying parameters. Hence with smart contracts, developers can build and deploy arbitrarily complex decentralized user-facing apps and services. DeLMS consist of several smart contracts such as Assignment.sol, Content.sol, File.sol, Message.sol etc.

A sample smart contract code enables the teacher to create and manage a repository to handle the assignment submissions from students is shown below.

```
 //SPDX-License-Identifier: Unlicense
pragma solidity ^0.8.0;

import "hardhat/console.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract NFTAssignment {
    string public name;
    address public owner;
    using Counters for Counters.Counter;
    Counters.Counter private _assignmentIds;
    Counters.Counter private _metadataIds;

    struct Assignment {
        uint256 assignmentId;
        string title;
        string assignmentInfo;
    }
    mapping(uint256 => Assignment) private
    idToAssignment;
    mapping(string => Assignment) private
    hashToAssignment;

    struct Metadata {
        uint256 id;
        uint256 assignmentId;
        string uri;
        address student;
        string result;
    }
    mapping(uint256 => Metadata) private
    idToMetadata;
    mapping(string => Metadata) private
    hashToMetadata;

    constructor(string memory _name) {
        console.log("Deploying metadata with name:",
     _name);
        name = _name;
        owner = msg.sender;
    }
    .......................................
}
```

This is a Solidity contract for an NFT (non-fungible token) assignment, where an assignment is represented as a struct that contains an ID, a title, and information about the assignment. The contract also includes a struct for metadata, which contains an ID, an assignment ID, a URI, a student address, and a result. The contract uses mappings to map IDs to assignments and metadata and also uses the Counters library from OpenZeppelin to generate unique IDs for assignments and metadata. The contract also has a constructor that sets the name and owner of the contract and logs the name when it is deployed.

Some important functions related to assignment.sol of smart contract is shown below:

```
   function updateName(string memory _name) public {
        name = _name;
    }
    function transferOwnership(address newOwner)
    public onlyOwner {
        owner = newOwner;
    }
    function fetchAssignment(string memory hash)
        public
        view
        returns (Assignment memory)
    {
        return hashToAssignment[hash];
    }
    function fetchMetadata(string memory hash)
        public
        view
        returns (Metadata memory)
    {
        return hashToMetadata[hash];
    }
    function createAssignment(string memory title,
    string memory hash) public {
        _assignmentIds.increment();
        uint256 assignmentId = _assignmentIds.
    current();
        Assignment storage assignment =
    idToAssignment[assignmentId];
        assignment.assignmentId = assignmentId;
        assignment.title = title;
        assignment.assignmentInfo = hash;
        hashToAssignment[hash] = assignment;
    }

    function createMetadata(uint256 assignmentId,
    string memory hash) public {
        _metadataIds.increment();
        uint256 metadataId = _metadataIds.current();
        Metadata storage metadata = idToMetadata[
    metadataId];
        metadata.id = metadataId;
        metadata.assignmentId = assignmentId;
        metadata.uri = hash;
```

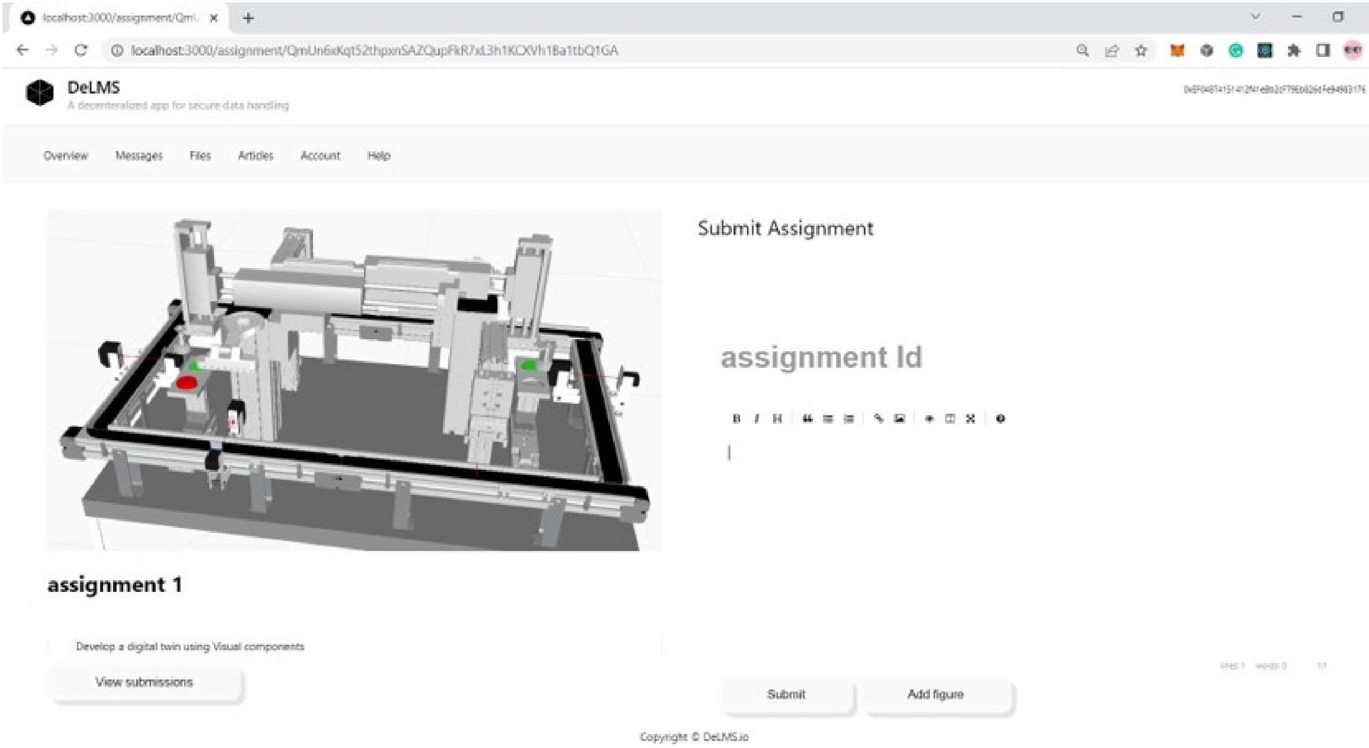

Fig. 5. DeLMS User Interface

```
        hashToMetadata[hash] = metadata;
        metadata.student = msg.sender;
    }
    function updateResult(uint256 metadataId, string
     memory hash)
        public
        onlyOwner
    {
        Metadata storage metadata = idToMetadata[
    metadataId];
        metadata.uri = hash;
        idToMetadata[metadataId] = metadata;
        hashToMetadata[hash] = metadata;
    }

    modifier onlyOwner() {
        require(msg.sender == owner);
        _;
    }
    modifier onlymetadataSender(uint256 metadataId)
    {
        Metadata storage metadata = idToMetadata[
    metadataId];
        require(msg.sender == metadata.student);
        _;
    }
}
```

It has several functions that allow different actions on the contract:

- The **updateName** function allows the owner to change the name of the contract.
- The **transferOwnership** function allows the owner to transfer ownership of the contract to another address.
- The **fetchAssignment** function allows anyone to retrieve an assignment by its hash.
- The **fetchMetadata** function allows anyone to retrieve metadata by its hash.
- The **createAssignment** function allows the owner to create an assignment with a title and hash.
- The **createMetadata** function allows anyone to create metadata with an assignment ID and hash and also saves the student's address that created it.
- The **updateResult** function allows the owner to update the metadata URI and also updates the hashToMetadata mapping.
- The **onlyOwner** and **onlymetadataSender** are modifiers that ensure that only the owner or the student who created the metadata can perform the action respectively.

The smart contract is written in solidity 0.8.0 version and it can be deployed on the Ethereum Virtual Machine. Initially, a smart contract is developed using the Hardhat (an Ethereum development environment) and tested in a personal Ethereum blockchain called Ganache - Truffle Suite and later deployed on Rinkeby Testnet and Polygon testnet which helps to explore and search the blockchain transaction. Once the testing phase is finished, We implemented a small laboratory-scale Ethereum private network consisting of Raspberry pi, beagle bone and PC nodes. These peers connected and formed a private network and then the smart contracts deployed on it. The same nodes are used for constructing the IPFS cluster and the data is stored in a decentralized fashion. The user interface for DeLMS is shown in the figure 5 and it has several functionalities like course management, submitting, evaluating and sharing educational content etc.

V. Conclusion and Future work

In conclusion, a decentralized learning management system (DeLMS), using Ethereum smart contracts and IPFS, provides a more secure and sustainable alternative to traditional LMS. DeLMS implements a secure, transparent, and decentralized platform for managing and delivering educational content that can revolutionize the way educational institutions operate.

By utilizing IPFS, DeLMS eliminates the need to constantly request a central server for the location of files, thereby accelerating file transfers and reducing energy consumption. This not only makes it more sustainable but also contributes to reducing the additional running and maintaining cost associated traditional LMS. In addition, smart contracts allow certain processes to be automated and ensure the integrity of student data and records.

However, as with any new technology, some challenges still have to be addressed. For example, the adoption of DeLMS may be resistant by educational institutions accustomed to traditional LMS and may find it difficult to switch to a new system. In addition, further research is needed to investigate the potential impact of DeLMS on various stakeholders such as students, teachers, and educational institutions.

Future work for DeLMS includes addressing these challenges and continuing to explore new use cases for the technology in the education sector. As the technology develops, we can expect to see more real-world implementations of DeLMS that can demonstrate its potential and benefits. Additionally, further research can be done to examine the scalability and interoperability of the proposed system and how it can be integrated into existing infrastructure.

The developed system exhibits novelty in its integration and utilization of Ethereum smart contracts and IPFS within the realm of educational content management, delivery, and student data integrity. It addresses limitations of traditional learning management systems, such as centralized control, lack of transparency, and high energy consumption, by leveraging the decentralized nature of blockchain technology and the distributed file system of IPFS. Future research can further enhance the system's novelty by focusing on challenges like scalability, interoperability, and exploring additional use cases and integrations within the educational infrastructure.

VI. Acknowledgements

This research was supported, in part, by the Kolarctic project Wood Industry 4.0 (grant number KO5023), and the Horizon Europe project Zero-SWARM funded by the European Commission (grant agreement: 101057083). The authors would like to acknowledge ChatGPT, the language model trained by OpenAI, for its contribution in assisting us with the writing of this article.

References

[1] N. A. Alias and A. M. Zainuddin, "Innovation for Better Teaching and Learning: Adopting the Learning Management System," 2005.

[2] "Sidney L Pressey." [Online]. Available: https://research.osu.edu/sidney-l-pressey

[3] N. Aldahwan and N. Alsaeed, "Use of artificial intelligent in learning management system (lms): A systematic literature review," *International Journal of Computer Applications*, vol. 175, pp. 975–8887, 08 2020.

[4] S. A. Raza, W. Qazi, K. A. Khan, and J. Salam, "Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model," *Journal of Educational Computing Research*, vol. 59, no. 2, pp. 183–208, Apr. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7509242/

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," p. 9.

[6] G. H L, A. Athreya, A. Kumar, A. Holla, S. Nagarajath, and R. Kumar V, "BLOCKCHAIN," May 2020, pp. 1–24.

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

[8] "What is Ethereum?" [Online]. Available: https://ethereum.org

[9] R. Sujeetha and C. A. S. Deiva Preetha, "A literature survey on smart contract testing and analysis for smart contract based blockchain application development," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 378–385.

[10] L. Ante, "Smart contracts on the blockchain – a bibliometric analysis and review," *Telematics and Informatics*, vol. 57, 10 2020.

[11] T. V. Doan, Y. Psaras, J. Ott, and V. Bajpai, "Towards decentralised cloud storage with ipfs: Opportunities, challenges, and future considerations," *IEEE Internet Computing*, pp. 1–10, 2022.

[12] J. Benet, "Ipfs - content addressed, versioned, p2p file system," 2014. [Online]. Available: https://arxiv.org/abs/1407.3561

[13] J. P. de Brito Gonçalves, G. Spelta, R. da Silva Villaça, and R. L. Gomes, "Iot data storage on a blockchain using smart contracts and ipfs," in *2022 IEEE International Conference on Blockchain (Blockchain)*, 2022, pp. 508–511.

[14] H. F. Ramadhan, F. A. Putra, and R. F. Sari, "News verification using ethereum smart contract and inter planetary file system (ipfs)," in *2021 13th International Conference on Information & Communication Technology and System (ICTS)*, 2021, pp. 96–100.

[15] S. K. Dwivedi, M. S. Obaidat, R. Amin, and S. Vollala, "Decentralized management of online user reviews with immutability using ipfs and ethereum blockchain," in *2022 International Mobile and Embedded Technology Conference (MECON)*, 2022, pp. 534–539.

[16] Y. Liu, Z. Wu, Y. Liu, K. Wang, W. Fan, and L. Lin, "Research on data sharing mechanism of power material supply chain based on blockchain," in *2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, 2021, pp. 345–350.

[17] X. Cheng and F. Qu, "Ocean data sharing based on blockchain," in *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, 2021, pp. 155–159.

[18] S. Lin, X. Wang, S. Nie, W. Kou, and J. Du, "Research on the sharing of equipment data based on blockchain," in *2021 6th International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2021, pp. 435–438.

[19] S. Badlani, T. Aditya, S. Maniar, and K. Devadkar, "Educrypto: Transforming education using blockchain," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2022, pp. 829–836.

[20] S. V, S. R, D. Mukhopadhyay, and S. K. Gupta, "Decentralized accreditation of educational attainments using blockchain," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1–4.

[21] A. Athreya, A. Kumar, S. Nagarajath, G. H L, R. Kumar V, S. D N, and R. K R, *Peer-to-Peer Distributed Storage Using InterPlanetary File System*, 01 2021, pp. 711–721.