Design of a Decentralised Autonomous Scientific Publishing Organisation

Sandi Gec*, Petar Kochovski†, Jovan Buragev‡ and Vlado Stankovski§
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
Email: *sandi.gec@fri.uni-lj.si, †petar.kochovski@fri.uni-lj.si, ‡jb5000@student.uni-lj.si, §vlado.stankovski@fri.uni-lj.si

***Abstract*—The scientific publishing process today is by many scientists still considered as non-optimal. Potential improvement areas include the overall publisher business models, the quality and transparency of the review processes, the replicability of scientific experiments, the reputation and informetrics, and other quantifiable and qualitative aspects. Decentralised Ledger Technologies (DLT) have already been used in the context of Decentralised Autonomous Organisations and may contribute to user identification, data and process verification and certification, reputation management and business models based on tokenomics principles. The following study presents the design and the implementation of a Decentralised Autonomous Scientific Publishing Organisation that operates under democratic principles and aims at improving the quality of the scientific publishing processes.**

***Keywords*—Blockchain, Smart Contract, Decentralised Autonomous Organisation, DAO**

I. Introduction

Nowadays there is a great interest for transparent processes and provable provenance especially in scientific field where may arise cases of plagiarism. Manuscripts are usually provided through a centralised system where different stakeholders plays a role in order to evaluate the manuscript. The main function of the traditional scientific publishing system is sharing novel research findings, concept and methodologies, thus, establishing communication with a wider audience over time, using various Web repositories. They act as an research intermediate between the readers and the author of the content. There is a believe that researchers follow all ethical criteria, for example that does not submit the same manuscript into different publishing entities (e.g. conferences, journals etc.). However, even the most widely reputed scientific publishers resort relays on (partially) centralised systems where the stakeholder credentials are fully centralised and differs among them. Moreover, the published content is stored in a centralised way, so that the publisher maintain full control over it. Our motivation is to design a novel democratic community driven scientific publishing process and thus providing the methodology of a Decentralised Autonomous Organisation (DAO). The pillar logic of the process will rely on on distributed ledger technologies that improves the content verification processes, transparently determines reputation of the stakeholders, and consequently, the consistency of the published content provided through fully distributed system where the decision of further upgrades are determined in democratic terms through governance and voting.

The goal of this study is to design an architecture of a novel DLT-based publishing system which provides several advanced functionalities such as improved functional components. A novel Proof of Truthful Publishing consensus protocol provided through dedicated Smart Contract, as a part of the architecture, will be the main contribution and innovation of this work. This novel protocol will improve the overall content verification quality. We will design a process of democratic selection of editors, which will improve the trust in the scientific publishing process in comparison with traditional centralised research publishing processes. Other key functionality components are the copyright claiming, publishing and reputation management components. They complement the overall medium design and enable efficient communication among components and collaboration among the stakeholders of the media industry. The copyright claiming and publishing components will improve the process in comparison with the traditional way due to the decentralisation approach, which is achieved through the use of DLT's including Smart Contracts. Furthermore, compared to the technologies used in traditional media, our approach is intended to improve the consistency of the content and the collaboration among the stakeholders of the medium by improving the trustworthiness of the published research.

The remainder of the paper is organised as follows. Section 2 covers our work in the context of other related works. Section 3 describes the existing scientific publishing use case. Section 4 depicts the potential upgrade of the scientific publishing use case process to a fully DAO. Section 5 presents preliminary results including Smart Contracts and section 6 draws the conclusions.

II. Related works

Fornes et al. [1] explain that a peer review may raise concerns about fairness, quality, performance, cost or accuracy. They particularly mention that the middlemen publishers can still impose partial policies and concentrate profits. Moreover, general publishing processes may still be improved, for example, by using content verification processes. Taylor et al. used DLT's [2] to build a news platform that combines news creation with decentralised networks as a means to delivering factual content, curated by a community of readers and writers. Po et al. [3] proposed another project that is using DLT's to securely track content metadata, such as timestamps, copyright, authorship, and distribution of digital assets. Its main purpose is to filter out false news by protecting the reputations of legitimate content providers. Civil [3] is a network of news organisations, journalists, and investors built using DLT's. However, all these projects are in very early stage. Steemit [4] is a popular and functional blockchain based public content platform that aims to support social media and online communities by returning much of its value to the people who provide valuable contributions by rewarding them with cryptocurrency. Blockchain technology has found its use also in the in the context of scientific publishing. For instance, EUREKA [5] utilises blockchain tokens to cover the cost for reviewing papers. ARTiFACTS [6] is another blockchain-based scientific publishing solution that provides proof-of-existence of early scientific work, allows asset creation, tracking and sharing. Similarly, Pluto [7] proposes a blockchain-based solution to exchange scientific data and know-how by using smart contracts and tokens to maintain copyright control. In contrast to these solution, this paper describes a publishing system that facilitates creation of distributed journals and a to more decentralised approach. Hence, our solution minimises central authorities and relays mainly on Smart Oracle services when accessing off-chain data. Moreover, it uses Self Sovereign Identification (SSI) [8] to enable privacy-preserving identification of the stakeholders in our system.

III. Review process in the Scientific Publishing use case

The review process in the domain of science follows a strict methodology that generally consist of two main stakeholders: (i)*author* that intends to publish novel scientific findings and (ii) *owner* that may be further elaborated as reviewer, editor or other related entities related to the type of the scientific manuscript (e.g. journal, scientific conference, workshop, symposium etc.). The conventional scientific use case (see Fig. 1) can be summarised as follows:

1) author submits the scientific manuscript,
2) editor (owner) performs a preliminary review of the manuscript and rejects the manuscript or assign the manuscript to the appropriate reviewers,
3) the reviewers evaluate the manuscript and communicate the decision (e.g. accept, to be further revised or reject) to the author,
4) if needed the authors iterate the manuscript by addressing the comments of the reviewers,
5) repeat the Step 3,
6) once the manuscript is accepted the publishing process starts.

Since the conventional use case consist of many stakeholder interactions it may be significantly improved not only to facilitate the publishing process but also to improve the transparency, availability of the software services by distributing them but it would also allow many scalable improvements such as introduction of the reputation process among the stakeholders, minimise the plagiarism, expand functionalities through the tokenization and many others. A general overview of the conventional scientific publishing use case with the potential gaps to address is illustrated in Figure 1. In the following section we describe the potential aspects that may benefit with the introduction of the DAO, Smart Contracts and other blockchain characteristics.

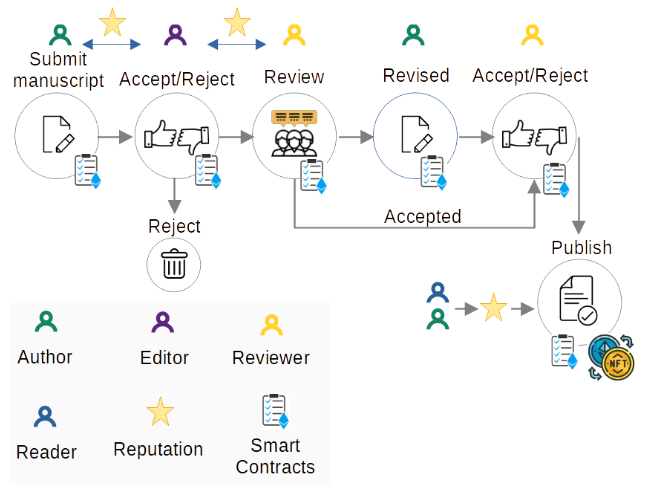

Fig. 1. General workflow of the Scientific Publication use case with identified potential gaps for Smart Contract integration

IV. Design of a Decentralised Autonomous Scientific Publishing Organisation

Scientific Publishing process consist of several central entities, from software services to publishing companies. The majority of these entities may become (partly) obsolete with the systematical integration of the DAOs. In general DAOs strive to provide an ecosystem that has the following properties:

- hierarchical organisation is replaced with flat democratised organisation,
- voting is required for any changes to be implemented,
- votes tallied, and outcome implemented automatically without trusted intermediary,
- offered services are handled automatically in a decentralised manner (e.g. through Smart Contracts) and
- all activities are transparent and mainly public.

By following the presented properties and other blockchain tokenomics benefits, we propose the key improvements in the Scientific publishing process.

A. High-level architecture

The design of a Scientific Publishing environment empowered with Blockchain technology relays on all involved entities interacting with the system. It is crucial to carefully design the workflow of the system pillar scenarios from which we built the architecture. First, we identified the following stakeholders and interactions among them: (1) author, (2) reader, (3) reviewer and (4) editor.

The general stakeholder workflow can be depicted as follows. In order for the publishers to verify and publish their records in our medium, they need to publish the content and claim copyright over it. Moreover, this feature is enabled through publishing and copyright claiming process. By introducing claim mechanisms we strive to minimise errors and improper use of the system. The readers is able to consume the verified content, including the ability to provide reputation to the verified content. The reputation process is covered by another component considered into the whole architecture. The editors act in the verification consensus protocol. As presented in the following sections, we analyses already existing verification consensus protocols and content verification methods. The goal here is to improve the traditional content verification protocols. The agents, trusted by the medium are chosen by some predefined criteria in a diversity and human-rights promoting, pluralistic and democratic way. The editors use predefined methods verifying the content and they are rewarded for their work. As an addition tightly overlapped with tokenomics, a fair and efficient reward system which is considered to be integrated backing complement and support the verification process using smart contracts and decentralised finance. This reward system may then be extended into a business model. Moreover, the system may be further elaborated to include open peer review to address other paradigms (e.g. plagiarism check, content relevance etc.).

All the identified functionalities are summarised in a comprehensive high-level architecture with four pillar components as illustrated in Figure 2. The key capabilities and features are further elaborated and described in the rest of the subsection.

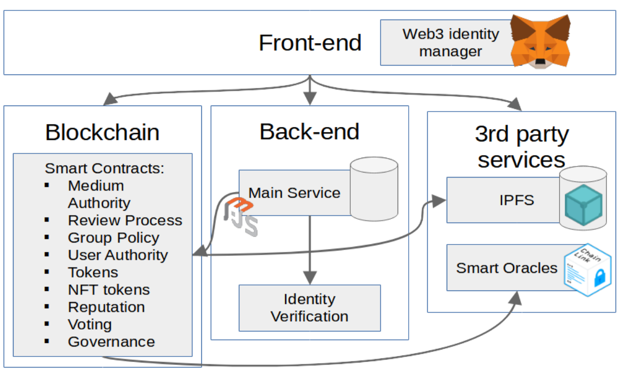

Fig. 2. High-level architecture indicates the main components and communication among them

B. Tokenomics

Cryptocurrency tokens are build upon a blockchain ledger and triggers operations with fees payed in native coins. Currently, one of the most popular Blockchain supporting tokens is Ethereum ledger. There are many standards for the tokens where in the context of tokenomics ERC-20 standard describing a fungible token, is the most suitable. Each token methodology is described in the Smart Contract - global/local token attributes, functions, function modifiers to restrict function access and other Smart Contract elements. Thus, the tokens should be designed not only to support the default token valutation and default ERC-20 operations but the process of the token design is tightly dependent or even overlapping on the business model of the token. Based on the latest findings in this field by P. Freni et al. [9], we should consider token behaviour based on morphological token classification covering three pillar classifier domains: (i) technology, (ii) behaviour and (iii) coordination.

Technological domain primarily focus on the support of an existing chain based on Ethereum Virtual Machine (EVM), fungible token representation (ERC-20), such as native Ethereum[1], Polygon[2], Build and Build (BNB) Smart Chain[3] or others that consist of significantly reduced transaction cost compared to Ethereum mainnet. Another crucial technological property following the DAO methodology is permissionless. In order to support horizontal scalability we strive to use a ledger technology that supports cross chain, that EVM actually does.

Behaviour domain covers different aspects basis on the abilities of the tokens. For example, it is possible to force the deflationary token behaviour with burn of certain amount of the tokens in specific actions (e.g. fees, actions such as minting of NFTs, periodical burns etc.). Moreover, the token economical characteristics should be defined before the token launch (e.g. spendability, expirability, fungibility, divisibility and tradability). Our token divisibility should be fractional to increase the maximum supply.

Coordination domain impact the actual business model functionalities since it defines various incentives. The supply strategy should be estimated based on the use case, in our case it should be several factors bigger than the research community in the world, although our stakeholders would be unique identified with the Non-Fungible Token (NFT) methodology described more in details in the following subsections. Practical functionalities should focus on discounts, revenues, rewards, dividends or earnings, reputation, governance and other similar incentives or stakeholder rights. For example, ordinary a review of a reputable reviewer has more value than a review from a reviewer with a minimal experience.

Our fundamental environmental functionalities rely on two tokens:

- ERC-20 fungible tokens needed to perform the majority of functionalities through Smart Contracts as described in the next Section.
- ERC-721 NFT collections for stakeholders minted in reached important milestones (e.g. after 100 written reviews etc.). The token specifics are described in the following Subsection.

C. Smart Contracts and on-chain data storage

Smart Contracts play a crucial role in our system since in theory the system is self sustainable because it fully rely on the selected EVM ledger. In the development process of the Smart Contracts we identified two main Contracts inherited into a

[1] https://ethereum.org
[2] https://polygon.technology/
[3] https://www.bnbchain.world/en/smartChain

single contract. The first Smart Contract involves the stakeholder overview and stores the users, user groups and supports various functionalities such as voting where the basic logic derives from OpenZeppelin[4] Contracts. The second derived Smart Contracts inherit the first one and further elaborate the entire logic behind the use case, all the interactions triggered by each stakeholder in the Scientific Publishing process (see Section V).

Storing data on-chain requires in-depth analysis of all the data that our environment. Qualifying the data based on the data types, frequency of the data (e.g. time-series data requires constant storing of big data chunks), besides the integrity and authenticity properties [10], needs to consider the privacy aspect by law such as General Data Protection Regulation (GDPR) in European Union. In order to preserve a high level of integrity all stakeholder based operations are executed through functions in the Smart Contracts where part of the data is stored off-chain through dedicated Smart Oracles or system Application Programming Interfaces (API). In addition it is possible to investigate science gateway perspective, especially on how to integrate research activities including the data with such within the publishing ecosystem.

D. Front-end

Front-end component is a Angular[5] Web page that supports the Smart Contract (on-chain) communication using web3js library[6] through a browser bridge MetaMask[7]. Other communication protocol is RESTful API covering off-chain interexchange of the data and access to 3rd party services. Since the wallets are owned by the stakeholders there is no need to login on the Web interface in a conventional manner (e.g. providing the e-mail and the password). Therefore, in the prototype design of the Smart Contracts a stakeholder should provide a wallet for each user role in order to authenticate accordingly to the preferred role as depicted in Figure 3.

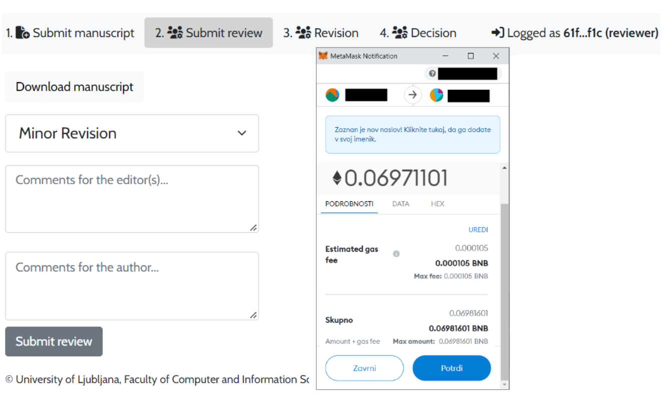

Fig. 3. Front-end example of a reviewer submitting the decision through a Smart Contract

[4]https://openzeppelin.com/
[5]https://angular.io/
[6]https://web3js.readthedocs.io/en/
[7]https://metamask.io/

E. Back-end Smart Contract setup

The backbone of the Smart Contracts is the back-end which enables EVM ledger interaction such as the deployment of the Smart Contract instances. Back-end integrates the framework Truffle Suite[8] which is used in both, the development process (e.g. preparation and execution of Solidity Unit tests) and production. The instances of the Smart Contracts are deployed through the back-end and simultaneously the off-chain data is stored in the the system database. The system centralised database stores just reference data of the Smart Contracts such as Contracts deployment addresses and other related metadata that is passed to the front-end component. The actual behaviour and specifics of the Smart Contracts are depicted in Section V.

F. Off-chain data storage

Storing data on-chain is not financially nor technically efficient. Therefore, we tend to store the metadata and the review process off-chain using a centralised database (or graph) or using a distributed database [11], [12]. The challenge here is to link the on-chain with the off-chain storage. DHT technology [13] can be used in order to achieve the aforementioned challenge as well as IPFS [14] for storing files in a decentralised way. The system will benefit in numerous ways such as faster interaction between stakeholders and lower costs.

G. Reputation mechanism and NFTs

The reputation mechanisms help stakeholders to decide whom to trust, they encourage trustworthy and discourage unwanted behaviour. Providing reputation to the published content, by the consumers, is levelling up the overall quality of the published articles and distinguish the content by its quality. Since the integrity is important, the reputation mechanism will be integrated in a decentralised way using the blockchain technology. A good example of a decentralised reputation system is the Reputable project[9].

H. NFT collections for the stakeholders

Stakeholders, involved in the proposed DAO based publishing process are primarily lead by research motivation as they all broaden their horizons. To further motivate the stakeholders, we propose a NFT collection distribution to the most devoted stakeholders when reached certain milestones. For example, after a reviewer provided certain amount of reviews (e.g. 100), the system mints an NFT collection and send it to the reviewer's wallet. Further, the NFT stored in the wallet may unlock additional functionalities or provide other benefits such as discounted prices. The benefits should be community driven through a democratic voting. This kind of blockchain based incentive should be provided to all types of stakeholders, event for readers while providing added value

[8]https://trufflesuite.com/
[9]https://ontochain.ngi.eu/content/reputable-provenance-aware-decentralized-reputation-system-blockchain-based-ecosystems

to the DAO system (e.g. providing scientific comments for published manuscripts).

V. Experimental study

In this section we provide analysis of proposed environment focused on Smart Contracts, mechanisms to comprehensively provide traceability, transparency and sustain the DAO methodology to minimise unnecessary central authorities. We first provide qualitative analysis between the existing process and our blockhain enhanced approach. Second, we propose preliminary Smart Contracts focusing on DAO based user roles and scientific publishing Smart Contract functionalities.

A. Qualitative assessment

The main workflow of the conventional and DAO based publishing processes consist of same main steps as depicted in Figure 1. Blockchain features are applied over the system implicitly by design; transparency, temper-proof data, availability, distribution. Other additional features, not available in existing publishing process solution [14], are NFT collectables, DAO user roles, voting, Smart Oracle service interaction and reputation mechanisms that are provided through the Smart Contracts. In the following subsection we provide prototype design of the Smart Contract with the description of our preliminary findings.

B. Smart contract design

The Smart Contracts are developed through a bottom-up approach methodology. The characteristics identified from the qualitative assessment is developed and packed into two Smart Contracts. First base Smart Contract defines the user roles of the system that is by design fully self sustainable. The derived Smart Contract (extends the base one) integrates the basic review process. Since the deployment operation is the most costly, we strive to deploy only one instance of the Smart Contract and thus minimise operational cost among the stakeholders and limit them only to the triggered functions as further described in the following subsections.

1) User roles: Conventional systems define users in (de)centralized databases managed by central authority or shared authorities. The aim of our study is to develop DAO based user roles that may be automatically added into the system for authors, readers and reviewers, or democratic governance in case of editors. The governance rules dictates that each active editor has the right to confirm a new editor member. In the initial setup the owner of the system has the voting power as defined in the constructor of the smart contract. Each new potential editor registers as editor through the call of the function *registerStakeholder('3')* where the status of the editor remains in *pending* till all the active editors including the Smart Contract owner does not approve the editor by triggering the function *confirmEditor('address of the pending editor')* where the status of the editor updates from *pending* to *active*. The logic of the workflow can be simply adopted by specific thresholds (e.g. 80% editors needs to approve a potential new editor) or other advanced functionalities (e.g. revoke single editor rights through governance). The presented workflow is described as a Smart Contract in Solidity language suitable for any EVM compliant ledger:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract StakeholdersDAO {
    mapping(address => Stakeholder) public
        stakeholder;
    address[] activeEditors; address public owner;

    struct Stakeholder {
        bool isValid; StakeholderGroup
            stakeholderGroup;
        StakeholderStatus stakeholderStatus;
        address[] awaitEditors; // other props
    }
    enum StakeholderGroup { Author, Reader,
        Reviewers, Editor }
    enum StakeholderStatus { Pending, Active,
        Deactivated }
    constructor() { owner = msg.sender;
        activeEditors.push(owner); }
    ...
    function registerStakeholder(uint _stg) public
        validStakeholder(_stg)
    {
        address _address = msg.sender;
        require(!stakeholder[_address].isValid, "the
            address is already registered");
        stakeholder[_address].isValid = true;
        // _stg stands for stakeholder group
        if (_stg == 1 || _stg == 2 || _stg == 3) {
            stakeholder[_address].stakeholderGroup =
                _stg == 1 ? StakeholderGroup.Author
                : StakeholderGroup.Reader;
            stakeholder[_address].stakeholderStatus
                = StakeholderStatus.Active;
            emit StakeholderIsRegistered(_address);
        }
        else {
            //create a copy of the active editors
            stakeholder[_address].awaitEditors =
                activeEditors;
            stakeholder[_address].stakeholderGroup =
                StakeholderGroup.Editor;
            stakeholder[_address].stakeholderStatus
                = StakeholderStatus.Pending;
            emit NotifyActiveEditors(_address);
        }
    }

    function confirmEditor(address _address) public
        onlyActiveEditors {
        bool found = false;
        for (uint8 i = 0; i < stakeholder[_address].
            awaitEditors.length; i++) {
            if (stakeholder[_address].awaitEditors[i
                ] == msg.sender) {
                delete stakeholder[_address].
                    awaitEditors[i];
                found = true;        break;
        }   }
        if (found && stakeholder[_address].
            awaitEditors.length == 0) {
            stakeholder[_address].stakeholderStatus
                = StakeholderStatus.Active;
            activeEditors.push(_address);
            emit NewEditorRegistered(_address);
        }
```

```
        else if (found) emit
            EditorApprovedByAnEditor(msg.sender,
            _address);
        else revert("invalid editor or already voted
            ");
    }
}
```

2) Scientific Publishing Contract: DAO based review process differs in conventional one by providing a fully transparent overview of the process. When an author submits a new manuscript the metadata is stored in IPFS dedicated database system where a Universal Unique Identifier (UUID) reference is generated. The submission function *manuscriptSubmission('UUID')* allows to store the author address, submission time and sets up the manuscript status to submitted. Each selected reviewer of the manuscript provides the review answer by triggering the function *review('UUID', 'decision status enum')*. The Smart Contracts can be further upgraded with Smart Oracle service which may serve to interact with IPFS system simultaneously while the presented functions are triggered.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

import "./StakeholdersDAO.sol";
contract ScientificPublishingProcess is
    StakeholdersDAO {
    mapping(string => Manuscript) public manuscript;
    uint maxEditors;     ...
    struct Manuscript {
        address author;
        uint256 submissionBlock;
        ManuscriptStatus manuscriptStatus;
        mapping(address => DecisionStatus)
            decisionStatus;
        ...
    }
    constructor(uint _me) StakeholdersDAO() {
        maxEditors = _me; }

    function manuscriptSubmission(string memory uuid
        ) public onlyAuthor {
        manuscript[uuid].author = msg.sender;
        manuscript[uuid].submissionBlock = block.
            timestamp;
        manuscript[uuid].manuscriptStatus =
            ManuscriptStatus.Submitted;
        ...         emit SuccessSubmission(uuid);
    }

    function review(string memory uuid,
        DecisionStatus decisionStatus) public
        onlyReviewer(uuid) {
        manuscript[uuid].decisionStatus[msg.sender]
            = decisionStatus;
        emit DecisionSubmitted(uuid, msg.sender,
            decisionStatus);
    }
}
```

VI. Conclusion

This paper presents a design and implementation of a blockchain-based decentralised and autonomous medium that operates under democratic principles and aims at improving the quality of the scientific publishing processes. The proposed solution improves the: content verification process, opinion polling and the transparency and truthfulness of the published content by introducing a novel Proof of Thruthful Publishing consensus protocol. In particular the protocol will, utilize especially tailored blockchain's smart contracts to anonymously select the editor and reviewers that will review the submitted manuscripts.

In our future research, we will design a reputation mechanism that will complement the selection of reviewers based on the quality of their prior reviewing work. Furthermore, additional evaluation of the proposed solution will be performed to additionally optimise the blockchain energy-efficiency, provide faster transaction rate, reduce transaction costs and allow intraledger transactions.

Acknowledgment

The research and development reported in this paper have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 957338 (ONTOCHAIN: Trusted, traceable and transparent ontological knowledge on blockchain) and from the Research Agency of the Republic of Slovenia under the research programme P2-0426 Digital Transformation for Smart Public Governance 1/1/22 - 12/31/27.

References

[1] A. Tenorio-Fornés, V. Jacynycz, D. Llop-Vila, A. A. Sánchez-Ruiz, and S. Hassan, "Towards a decentralized process for scientific publication and peer review using blockchain and IPFS," in *52nd Hawaii International Conference on System Sciences, HICSS 2019, Grand Wailea, Maui, Hawaii, USA, January 8-11, 2019*, T. Bui, Ed. ScholarSpace, 2019, pp. 1–10. [Online]. Available: http://hdl.handle.net/10125/59901

[2] S. Singh and D. Taylor, "Decentralized news network," *(2018)*. [Online]. Available: https://www.academia.edu/38070328

[3] I. Bernat, "Blockchain in journalism," *Tow Center for Digital Journalism, Columbia University (2019)*. [Online]. Available: https://doi.org/10.7916/d8-5jm6-w328

[4] C. Li and B. Palanisamy, "Incentivized blockchain-based social media platforms: A case study of steemit," *Association for Computing Machinery*, p. 145–154, 2019.

[5] A. Schaufelbühl, S. R. Niya, L. Pelloni, S. Wullschleger, T. Bocek, L. Rajendran, and B. Stiller, "Eureka – a minimal operational prototype of a blockchain-based rating and publishing system," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019, pp. 13–14.

[6] "Harnessing blockchain for real-time recognition of scientific and academic research," Blockchain-Based Platform for Scientific and Scholarly Research, 2022. [Online]. Available: https://artifacts.ai/researcher-blog/

[7] "Breaking down the barriers in academia," 2018. [Online]. Available: https://assets.pluto.network/Pluto_white_paper_v04_180719_1355_BSH.pdf

[8] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103 059–103 079, 2019.

[9] P. Freni, E. Ferro, and R. Moncada, "Tokenomics and blockchain tokens: A design-oriented morphological framework," *Blockchain: Research and Applications*, vol. 3, no. 1, p. 100069, 2022.

[10] T. Hepp, M. Sharinghousen, P. Ehret, A. Schoenhals, and B. Gipp.

[11] T. M. et al., "Bigchaindb: A scalable blockchain database," *In: BigchainDB (2016), pp. 1–65*. [Online]. Available: https://www.academia.edu/26845466

[12] X. L. Yu, X. Xu, and B. Liu, "Ethdrive: A peer-to-peer data storage with provenance," 06 2017.

[13] H. Z. W. X. Yu, "Distributed hash table," 2013.

[14] J. Benet, "IPFS - content addressed, versioned, P2P file system," *CoRR*, vol. abs/1407.3561, 2014. [Online]. Available: http://arxiv.org/abs/1407.3561