Towards an Approach to Organization of Decentralized Business Process Model Repository

Andrii Kopp, Dmytro Orlovskyi and Sergey Orekhov

National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61002, Ukraine

Abstract

This paper proposes an approach to the organization of a decentralized business process model repository to provide a secure and stable software solution for keeping and accessing business process model collections. State-of-the-art studies consider the problem of managing large collections of business process models as an extremely complex challenge, whereas the security and integrity of such corporate knowledge assets are crucial features of repository software. Today blockchain technologies are used not only as ledgers of financial transactions, but also as general-purpose distributed databases that provide high-level security, integrity, and availability. Thus, organizations may benefit from using blockchain platforms and knowledge-sharing protocols built on top of them. Therefore, in this paper, we consider the problem of decentralized blockchain-based business process model repository development using smart contract and decentralized application technologies. Presented software prototype of limited functionality is developed on top of the Ethereum test network, accessible using the single page Web3 application. Developed software prototype is validated, obtained results are discussed, the conclusion is made, and the future work is formulated.

Keywords [1]

Business Process Model, Blockchain, Repository, Knowledge Sharing, Smart Contract

1. Introduction: Related Work and Problem Statement

The current trend of the digital transformation encourages large enterprises, medium-sized companies, and even small businesses to focus on detection, analysis, and improvement of their business processes by deploying BPM (Business Process Management) suites that automate routine activities. Such an approach is known as BPM, while described cycle of continuous process improvement is referred to as BPM lifecycle. Business processes are considered as structured sequences of activities performed by employees and other stakeholders to transform raw information or materials into products or services valuable for customers, either external or internal. For example, a sick leave application is processed to satisfy an internal customer – an employee who requested the sick leave, while order processing serves an external customer – a company's client or a counter-party organization, which requires order fulfillment. Each of such business scenarios is called a business process and, traditionally for BPM projects, is represented using graphical models (similar to flowcharts used to describe algorithms) to capture, store, and share knowledge about organizational activities. Captured knowledge about ongoing business processes represented in the form of graphical schemes could be used to train new employees (future process participants) or to detect inefficiencies in workflows to improve organizational activities through business process automation (replacing manual routine tasks with scripts) or re-engineering (rebuilding whole business process scenarios from scratch).

Thus, it is natural for big enterprises to have extremely large collections of hundreds or even thousand [1] of business process models. Keeping and accessing such volumes of business process models could be possible with the use of enterprise-level techniques and software solutions. Since availability, integrity, and security are crucial features for enterprise collections of business process

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine

EMAIL: kopp93@gmail.com (A. Kopp); orlovskyi.dm@gmail.com (D. Orlovskyi); sergey.v.orekhov@gmail.com (S. Orekhov)
ORCID: 0000-0002-3189-5623 (A. Kopp); 0000-0002-8261-2988 (D. Orlovskyi); 0000-0002-5040-5861 (S. Orekhov)

models, in this paper we consider the usage of blockchain technology to keep and manage business process models securely and stably. The research subject is the collection of business process models organized using blockchain technology. The research objective includes an approach to managing the blockchain-based storage of business process models.

This paper is structured in the following way: current section outlines the introduction, as well as the state-of-the-art overview (sub-section 1.1) and problem statement (sub-section 1.2); section 2 demonstrates an approach to the organization of a blockchain-based collection of business process models, while Section 3 introduces developed software solution together with a discussion of its usage results; Section 4 contains conclusion and planned directions of future research in this field.

1.1. Related Work

1.1.1. Business Process Model Repository

According to Yan et al. [1], managing enormous collections of business process models is a complex problem indeed, which requires special software tools to store, search, and manage business process model versions [1]. Such software was called a "business process model repository" [1] and, according to Elias [2], it should correspond to particular requirements, e.g. support of a standard business process modeling notation BPMN (Business Process Model and Notation) [3], displaying of both graphical and textual process descriptions, provision of multiple model versions for the same process, model search and categorization, model analysis and comparison, support of relationships between business process models, etc. However, there are external access and security [2] requirements, which fulfillment is critical for corporate assets. Access control and integrity control features were also mentioned by La Rosa et al. [4], as capabilities of an advanced business process model repository software named "APROMORE" [4]. Also, access, integrity, and, in addition, transaction management features were mentioned as part of the framework for business process model repositories proposed by Yan and Grefen [5]. Researches made in this period are devoted to efficient querying of business process models, stored in a repository, using graph-based data structures and indexing approaches similar to search engines [6, 7], and comparison of business process models using various similarity measures based on business process structure and semantic [8, 9]. Nevertheless, all of these studies have mentioned accessibility and security as repository features, but not as their main research objectives.

1.1.2. Blockchain-based Business Process Management

First studies in this direction were published in [10] and [11] and later in [12] by López-Pintado et al., who presented the blockchain-based business process management system named "Caterpillar". This system captures process instances in the Ethereum blockchain by translating BPMN models into Solidity smart contracts that drive business process execution. In details translation of BPMN models into smart contracts is described in paper [13] published by the same group of researchers. Collaborative execution of business processes supported by a blockchain platform and smart contracts was also considered in [14]. As the extension of the Caterpillar blockchain-based BPM system, authors of [15] have proposed an approach to a process mining of event logs produced by business processes, in which execution data is stored on the blockchain [15].

Shared business process modeling and versioning rather than execution was proposed by Härer [16]. According to the proposed approach, public and private business process models could be created, managed, and shared among the decentralized participants in a transactional manner [16]. Then Fill and Härer extended their idea into the concept of "Knowledge Blockchain" where enterprise models (in [17] they also gave examples on BPMN process models) could be kept in the immutable and tamper-resistant way with authorship and ownership proofs [17].

Another paper focused on blockchain-based business process management considers cross-organizational business processes, which include collaborative activities distributed among businesses and individuals, based on blockchain technology [18]. In [19] Viriyasitavat and Hoonsopon proposed a blockchain-based architecture for consensus in collaborative business processes [19]. One more research in the field of cross-organizational blockchain-based business processes considers the integration of IT (information technology) systems organizations use to support business process execution and monitoring [20]. Another research by Milani and Garcia-Banuelos considers inter-organizational business processes executed in a blockchain-based ecosystem with smart contracts used both as process executioners and repositories of process data [21].

1.1.3. Blockchain Platforms

Since the crypto-boom of 2017, blockchain technology went far beyond the Bitcoin cryptocurrency, even though it was initially designed to fit digital currency requirements, such as forgery resistance, immutability, and decentralized community-backed governance [22]. Initially created to implement cryptocurrencies, the blockchain technology works with transactions (e.g. containing recipient's address, number of coins to be sent, and sender's signature), which are consolidated into blocks that contain hash values of blocks generated before, by which creating a chain of irreversible and immutable blocks [23]. Therefore, data authenticity and consistency could be proven by checking the conformity of hash values back to the initial block of the whole blockchain [24]. Thus, blockchain transactions do not need a "trusted entity" for processing, they could be executed fast and at a low cost, cannot be altered, and could be easily traced [24].

Such benefits could not be unnoticed by industry and to support business process execution on top of blockchains, "smart contracts" were introduced as the computer programs that run in the blockchain platform and record results of their execution as an immutable transaction into the blockchain [24]. Being a blockchain platform, Ethereum represents a peer-to-peer network of nodes that maintain a distributed ledger of transactions, whereas its main purpose is a "world computer" that runs smart contracts as general-purpose computer programs created using Solidity language similar to JavaScript by its syntax [25]. Unlike Bitcoin, Ethereum and other platforms that support smart contracts are called "programmable blockchains", which are used to create decentralized applications or "DApps" [25].

DApps use smart contracts as back-end code and blockchains as databases in contrast to traditional applications backed by centralized servers, whereas the frontend of DApps is usually created with the traditional combination of HTML (HyperText Markup Language), CSS (Cascading Style Sheets), and JavaScript together with the "web3.js" library is used to access Ethereum API (Application Programming Interface) [25]. By April 2021, five leading smart contract platforms could be used as "programmable blockchains": Ethereum, Polkadot, Solana, EOS, and Binance Smart Chain, however, Ethereum is still the major player and its future development may cement it as the dominant smart contract platform once and for all [26].

1.2. Problem Statement

Considering the relatively young age of blockchain-based business process management, there are mostly proof-of-concept or experimental solutions currently existing (i.e. Caterpillar and others [10 – 15]). These solutions consider the cross-organizational performance of business processes supported on top of blockchain platforms. However, to our best knowledge, the enterprise knowledge sharing approach using a blockchain platform and the BPMN process modeling standard were not proposed yet. Hence, using programmable blockchain platforms, such as Ethereum, there could be created a decentralized application to store and manage collections of business process models as part of the inter-organizational repository of corporate knowledge (see Figure 1).

According to the conceptual model shown in Figure 1, a blockchain-based repository of BPMN models could be implemented as multiple smart contracts: a main smart contract, which addresses one or several smart contracts that represent collections of business process models. As the frontend should be created a decentralized web application that will get the registry of process model collection from the main smart contract and access respective collections of BPMN models. Therefore, such a repository may contain multiple business process models collections owned by different organizations or individuals supporting this enterprise knowledge sharing initiative.

Financial capabilities of modern blockchain platforms (i.e. cryptocurrencies and custom tokens) could be used in the repository to exchange BPMN models of best-practice business processes on a commercial basis (e.g. access to certain collections could be granted as per subscription or one-time purchase). However, the initial goal is to define the generic structure and behavior of a business process model collection built on top of a blockchain platform. Then, a proof-of-concept smart contract and corresponding DApp should be implemented, e.g. using the Ethereum platform as the most popular one.

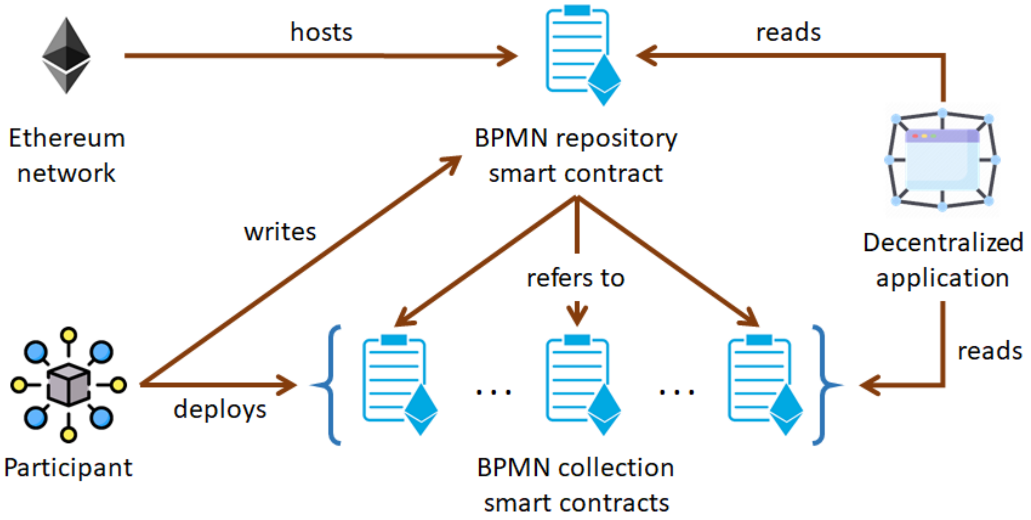

hosts
reads
Ethereum network
BPMN repository smart contract
writes
Decentralized application
refers to
reads
Participant
deploys
...
...
BPMN collection smart contracts

Figure 1: Conceptual model of a blockchain-based business process model repository

2. Proposed Approach to Organization of a Blockchain-based Business Process Model Collection

2.1. Proposed Formal Definitions of a Blockchain-based Business Process Model Collection

Business process model collection built on top of a certain blockchain platform could be described using the following formal definitions.

2.1.1. Definition 1. Collection of Business Process Models

Collection of business process models is the set of documents prepared in BPMN 2.0 format (XML-based files), which represent organizational activities for knowledge gathering and sharing purposes. It could be formally described using the following tuple:

$$C = \langle own, MR, m, P, A \rangle. \tag{1}$$

where:

— own is the owner of a collection of business process models (in blockchain platforms it could be defined by the address, e.g. as the hexadecimal string);

— MR is the list of records about business process models associated with this collection;

— m is the number of business process model records in the list MR (i.e. how many business process models are referenced in this collection);

— P is the permissions mapping, which defines users allowed to access this collection (in blockchain platforms users also could be defined by their addresses, as well as the collection owner own);

— A is the set of algorithms used to manage a collection of business process models.

2.1.2. Definition 2. Business Process Model Record

Business process model record is the structure that consists of business process model attributes, which could be described using the following tuple:

$$MR_i = \langle t_i, u_i, h_i, desc_i, ind_i, ts_i \rangle, i = \overline{1,m}. \quad (2)$$

where:

- t_i is the title of a business process model referenced in this record;
- u_i is the direct URL (Uniform Resource Locator) used to access a BPMN 2.0 document that corresponds to a referenced business process model;
- h_i is the hash value of a BPMN 2.0 content (XML-based file) that describes a business process (that could be calculated using secure hash algorithms, i.e. SHA256 or SHA512, or others); hash value is used to check the authenticity of a business process model referenced in a collection;
- $desc_i$ is the annotation or brief description of a business process model referenced in this record;
- ind_i is the industry name to which a business process described by a referenced in this record model belongs;
- ts_i is the timestamp that shows when this record about a business process model was made to the blockchain.

In fact, according to the proposed approach business process models are stored elsewhere (e.g. on shared file servers or version control system servers that support collaborative work, such as GitHub or GitLab), while only URLs and hash values of BPMN models are stored in the blockchain. This method is the most efficient: it saves a huge amount of space and cost, while making a collection tamper-traceable [27].

2.1.3. Definition 3. Permissions Mapping

Permissions mapping is the function that associates each address of a blockchain network to the binary value, which determines whether such address is allowed to add new business process models into this collection (1) or not:

$$P : addr \to \{0,1\}, addr \in Addr. \quad (3)$$

where:

- $addr$ is the address in a blockchain network, which belongs to a set of all addresses in the network $addr \in Addr$; for the owner's address $P(own) = 1$, $own \in Addr$;
- values of the codomain $\{0,1\}$ of P determine whether a user with network address $addr$ is allowed to add new business process models to a collection, $P(addr) = 1$, or not $P(addr) = 0$.

Besides the owner's address $own \in Addr$, $P(own) = 1$, by default for all of the addresses existing in a blockchain network $P(addr) = 0$, $addr \in Addr$, unless another is given using one of the algorithms for managing a collection of business process models.

2.1.4. Definition 4. Algorithms to Manage a Collection of Business Process Models

Algorithms to manage a collection of business process models is the set of algorithms used to implement certain operations with the collection of business process models, including adding the reference to a BPMN model and setting access permissions. In general, such algorithms could be represented as the following tuple:

$$A = \langle add, set \rangle. \quad (4)$$

where:

- add is the algorithm of adding new record about a business process model referenced (2) in this collection (1);
- set is the algorithm of setting access permissions (either granting or revoking) for a given user's address (3).

Considered algorithms (4) used to manage the collection of business process models will be described in detail in further subsection.

2.2. Proposed Algorithms of a Blockchain-based Business Process Model Collection

In the subsection will be outlined in details algorithms (4) for managing the collection of business process models (1). Each of algorithms add and set are given below.

2.2.1. Algorithm of Adding New Record About a Business Process Model

It is denoted as add (4) and takes as input:

— title of a referenced business process model, t_{new};

— direct URL link to access a BPMN 2.0 document, u_{new};

— hash value of a BPMN 2.0 file used to check the authenticity of a referenced business process model, h_{new};

— annotation or brief description of a referenced business process model, $desc_i$;

— industry name to which a described business process belongs, ind_i;

— timestamp that shows when a referenced business process model was added, ts_i.

This algorithm add consists of following steps:

Step 1. Get the sender's address (i.e. a blockchain address of a user who initiated the execution of this algorithm) $send \in Addr$.

Step 2. Determine the access permission $P(send)$ given to the obtained sender's address $send \in Addr$. Check whether the sender's address is given with the permission (3) to access a collection of business process models, i.e. if $P(send) = 1$, $send \in Addr$, then proceed to Step 3. Otherwise, if $P(send) = 0$, execution of the algorithm should be finished.

Step 3. Create a new record about a business process model that should be added to the blockchain $MR_{new} = \langle t_{new}, u_{new}, h_{new}, desc_{new}, ind_{new}, ts_{new} \rangle$, where $new = m+1$.

Step 4. Add created record MR_{new} to the end of the list of business model records MR (2).

Step 5. Increase the number of business process model records, $m = m+1$. Finish the algorithm.

The sender's address $send \in Addr$ could be obtained from a transaction fired by a certain blockchain platform user. According to the outlined algorithm, only the collection's owner $own \in Addr$ should be capable of adding new business process model references.

Later, when making collections of business process models accessible on a commercial basis, the proposed algorithm could be modified to allow non-owners to publish their models into collections for rewards in native repository tokens (quasi-cryptocurrency) that could be then spent to access private business process model collections.

2.2.2. Algorithm of Setting Access Permissions

It is denoted as set (4) and takes as input:

— user's address in a blockchain platform, $user \in Addr$;

— access granting or revoking value – 0 or 1 (in the smart contract implementation we may use "true" or "false" Boolean constants), $access$.

This algorithm set consists of following steps:

Step 1. Get the sender's address $send \in Addr$.

Step 2. Compare obtained sender's address $send \in Addr$ to the collection owner's address $own \in Addr$. If compared addresses are equal, then proceed to Step 3 and finish.

Step 3. Compare the user’s address $user \in Addr$ to the collection owner’s address $own \in Addr$. If compared addresses are not equal, then proceed to Step 4 and finish.

Step 4. Set access permission for the given user, $P(user) = access,\ user \in Addr$.

According to the outlined algorithm, only the collection’s owner $own \in Addr$ should be capable of setting access permissions. Moreover, the owner cannot manage its own access permission, i.e. $user \neq own$ condition should be met.

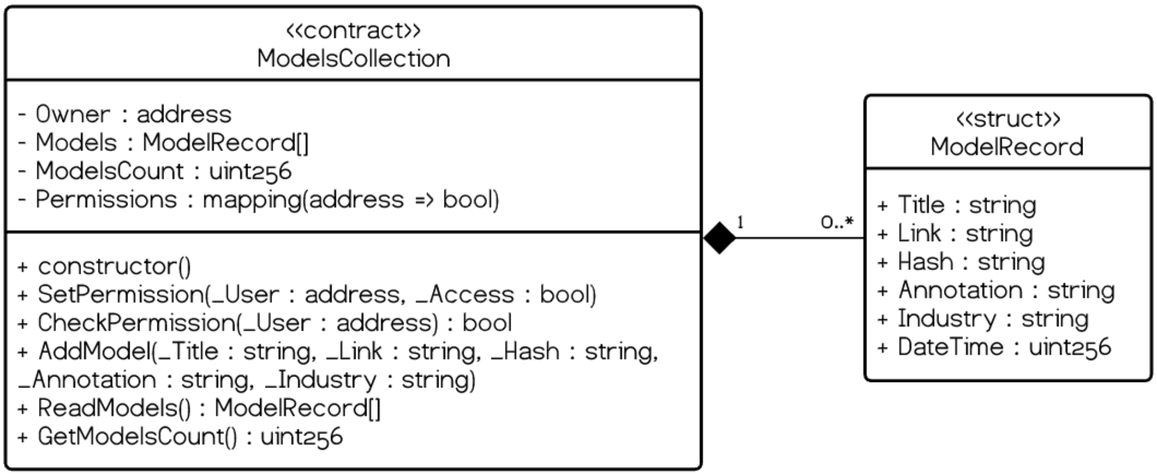
3. Results and Discussion

3.1. Development of an Ethereum Smart Contract Prototype

We have selected the Ethereum blockchain platform for smart contract implementation because of its dominance in the area [26], Turing complete programming language Solidity [25], and availability of development tools:

- Remix IDE (Integrated Development Environment) for smart contract programming and debugging;
- Ropsten test network (with the Ethereum cryptocurrency of no value, available from so-called “faucets” for development purposes [28]), which is the proof-of-work (transactions are confirmed by “miners” who lend their computing power for cryptocurrency rewards) and, thus, the best representation of the main network;
- MetaMask wallet for interaction with the Ethereum network (i.e. transaction sending, balance management, etc.).

In terms of a UML (Unified Modeling Language) class diagram notation, the static structure of a developed smart contract prototype could be shown as follows (see Figure 2).



<<contract>>
ModelsCollection
- Owner : address
- Models : ModelRecord[]
- ModelsCount : uint256
- Permissions : mapping(address => bool)
+ constructor()
+ SetPermission(_User : address, _Access : bool)
+ CheckPermission(_User : address) : bool
+ AddModel(_Title : string, _Link : string, _Hash : string, _Annotation : string, _Industry : string)
+ ReadModels() : ModelRecord[]
+ GetModelsCount() : uint256
1
0..*
<<struct>>
ModelRecord
+ Title : string
+ Link : string
+ Hash : string
+ Annotation : string
+ Industry : string
+ DateTime : uint256

Figure 2: Structure of the smart contract prototype

This “ModelsCollection” smart contract contains several private fields, which correspond to definition (1):

- “Owner” is the address of a user who deployed this contract to the blockchain;
- “Models” is the array of business process model records described by the respective structure “ModelRecord” concerning definition (2);
- “ModelsCount” is the number of referenced business process models and increments when a new model is recorded to the collection;
- “Permissions” is the mapping between user addresses and given or revoked permissions, according to definition (3).

Methods of the smart contract correspond to the algorithms defined in (4). Besides the constructor, which initializes the “Owner” field and grants permissions to the owner respectively, the remaining methods implement considered algorithms:

- “AddModel” takes input parameters that describe a BPMN model and makes a record to the array “Models” (according to Fig. 2);
- “ReadModels” checks permissions of a user who called this method and returns collection of recorded business process model references (according to Fig. 3);
- “SetPermission” allows the owner to grant or revoke access permissions to or from the mentioned user address (according to Fig. 4).
- “CheckPermission” allows any user to check their permissions.

The source code of the developed smart contract is available at [29]. In the current implementation, the constructor also creates four records of sample BPMN models: “Dispatch of goods”, “Insurance recourse”, “Credit scoring”, and “Self-service restaurant” taken from a public GitHub repository provided by Camunda for research purposes [30].

3.2. Development of a Web Application Prototype

Created web application prototype shows a decentralized application that uses HTML, CSS, and JavaScript together with the “web3.js” library to work with the smart contract demonstrated in the previous sub-section. Its source code is available at [29].

The system architecture of the proposed solution could be demonstrated using the UML deployment diagram (see Figure 3).

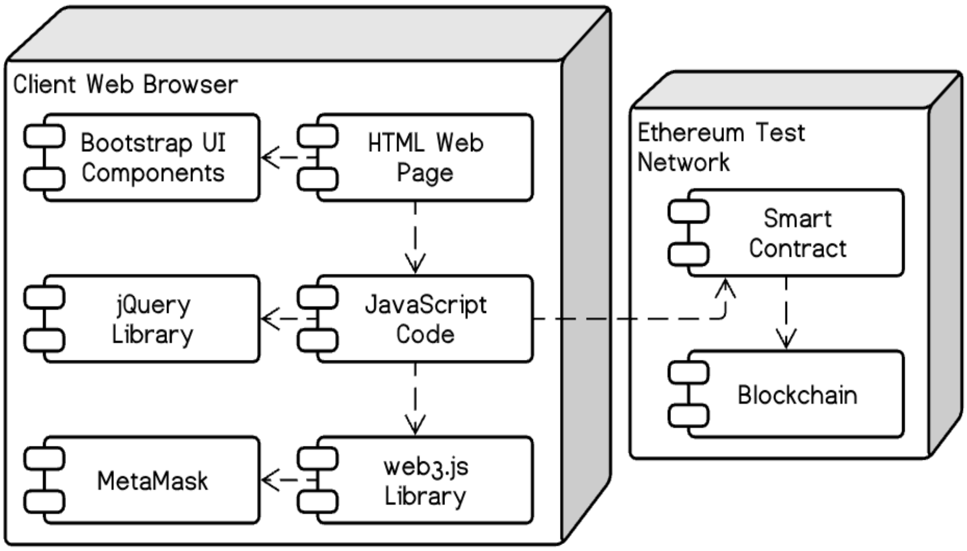
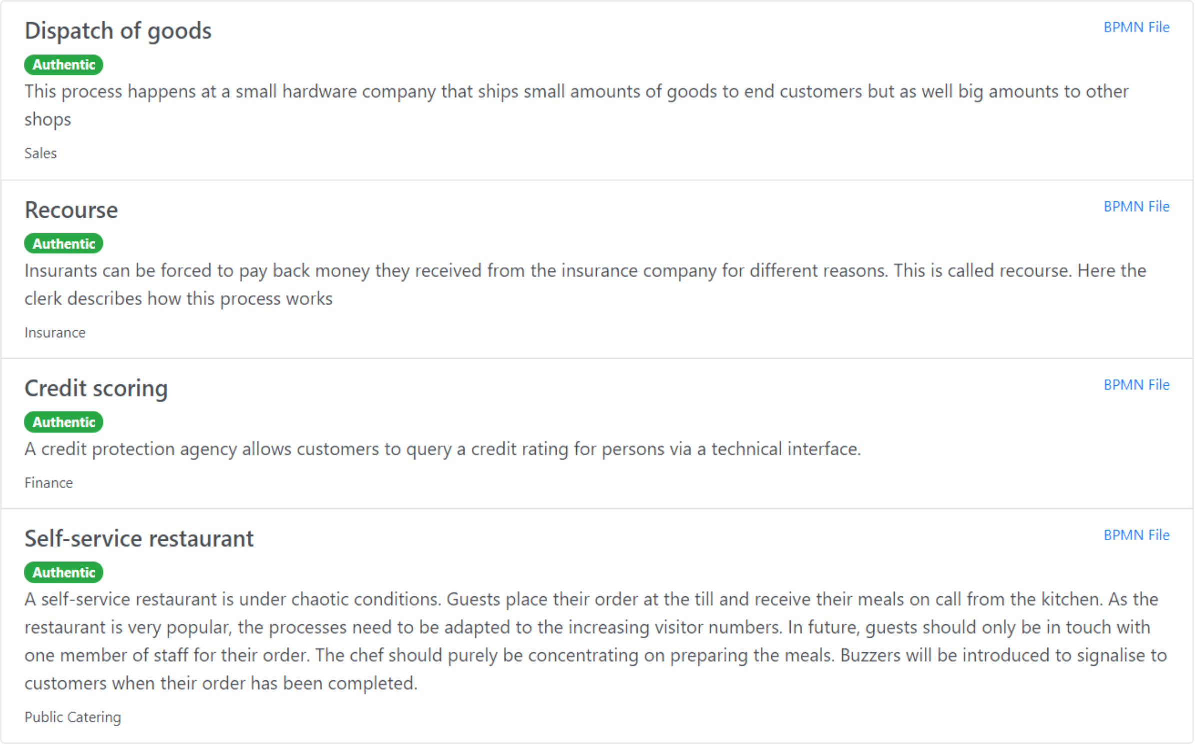


Figure 3: The system architecture of the web application prototype

When users try to access the business process model collection provided by the smart contract, their Ethereum addresses are taken from the MetaMask (users should be preliminarily authorized and connected) or another wallet, access permissions are checked, and the list of reference BPMN models is displayed using the web page (see Figure 4).

All of these procedures are implemented as the JavaScript code powered by web3.js and jQuery and libraries to read the smart contract and show results respectively. In order to check the authenticity of a business process model (whether it was altered), the CryptoJS library is used to calculate the SHA256 hash value of a BPMN document requested using the stored link, which is then compared to the model’s hash value stored in the blockchain. Model’s authenticity could be checked when accessing the collection through a web application (see Figure 4).

As it is shown in Fig. 8, a business process model could be downloaded as the BPMN file for its further usage. For now, this is all functionality of the prototype DApp created to work with the collection of business process models implemented using the Ethereum smart contract. When the owner needs to add new models to the collection, add, or change given user permissions, the smart contract could be used directly from the network explorer in the reading or writing modes [31].



Dispatch of goods

This process happens at a small hardware company that ships small amounts of goods to end customers but as well big amounts to other shops

Sales

Recourse

Insurants can be forced to pay back money they received from the insurance company for different reasons. This is called recourse. Here the clerk describes how this process works

Insurance

Credit scoring

A credit protection agency allows customers to query a credit rating for persons via a technical interface.

Finance

Self-service restaurant

A self-service restaurant is under chaotic conditions. Guests place their order at the till and receive their meals on call from the kitchen. As the restaurant is very popular, the processes need to be adapted to the increasing visitor numbers. In future, guests should only be in touch with one member of staff for their order. The chef should purely be concentrating on preparing the meals. Buzzers will be introduced to signalise to customers when their order has been completed.

Public Catering

Figure 4: Business process models requested from the smart contract

3.3. Validation of a Decentralized Application Prototype

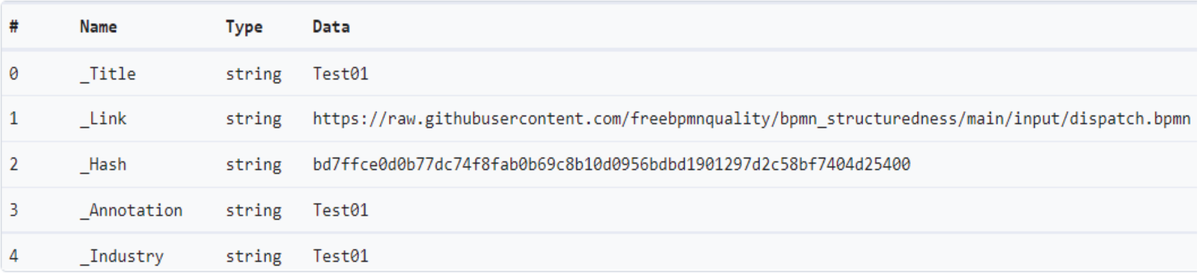
To validate the developed software prototype we used a BPMN model of the "Dispatch of goods" business process from [30]. This model was recorded to the blockchain using a smart contract interface provided at [31]. It is available at [32], while its hash is:

— *"fb96d72d6c03821022ba871c1727fc124067c502892c07bfc5fe1d6b99639282".*

This model was added to the collection by contract's owner, whose Ethereum wallet address is:

— *"0xC0753Fff03d88B34B9538d0A93D64831EE85d95C".*

The "Dispatch of goods" model was published by calling the "AddModel" method, corresponding transaction details are shown in Figure 5.



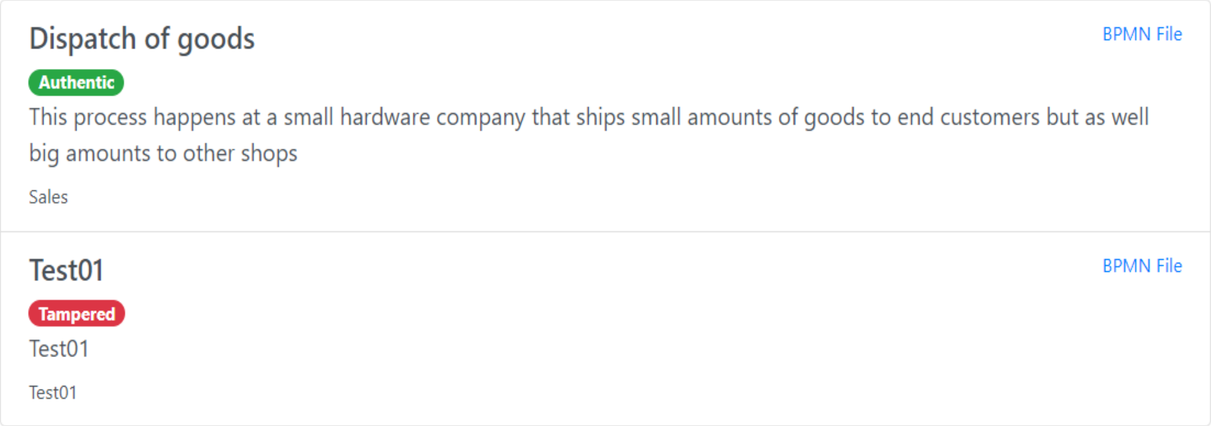
#	Name	Type	Data
0	_Title	string	Test01
1	_Link	string	https://raw.githubusercontent.com/freebpmnquality/bpmn_structuredness/main/input/dispatch.bpmn
2	_Hash	string	bd7ffce0d0b77dc74f8fab0b69c8b10d0956bdbd1901297d2c58bf7404d25400
3	_Annotation	string	Test01
4	_Industry	string	Test01

Figure 5: Details of a business process model publishing transaction

This transaction has the following ID, which can be used to check it in the Ropsten Etherscan:

— *"0xfb5d67e6b0350df49960b2d52de2c2a632e91079053c919b2b43f8e2f0845d3c".*

Let us consider a case when a business process model [32] was altered after was recorded to the blockchain. Even a small change of a model leads to the new hash value and the negative result when checking the model's authenticity (see Figure 6).



Dispatch of goods

Authentic

This process happens at a small hardware company that ships small amounts of goods to end customers but as well big amounts to other shops

Sales

BPMN File

Test01

Tampered

Test01

Test01

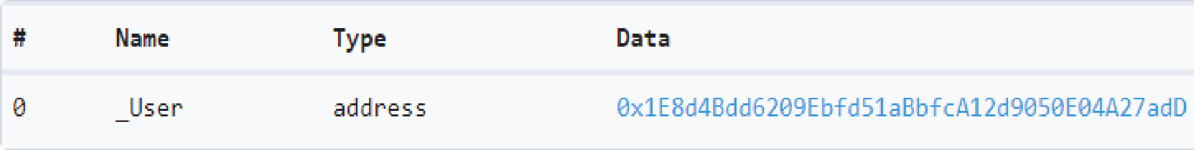
BPMN File

Figure 6: Example of tampered business process model detection

As it is shown in Figure 6, the recently added business process model (see Figure 5) was accompanied by the wrong hash value, so the DApp displays it as tampered. The problem is that the given hash value does not equal the calculated SHA256 value:

- *"bd7ffce0d0b77dc74f8fab0b69c8b10d0956bdbd1901297d2c58bf7404d25400" (given);*
- *"fb96d72d6c03821022ba871c1727fc124067c502892c07bfc5fe1d6b99639282" (actual).*

Then let us grant user "0x1E8d4Bdd6209Ebfd51aBbfcA12d9050E04A27adD" with the publishing permission by using the "SetPermission" method (see Figure 7).



#	Name	Type	Data
0	_User	address	0x1E8d4Bdd6209Ebfd51aBbfcA12d9050E04A27adD

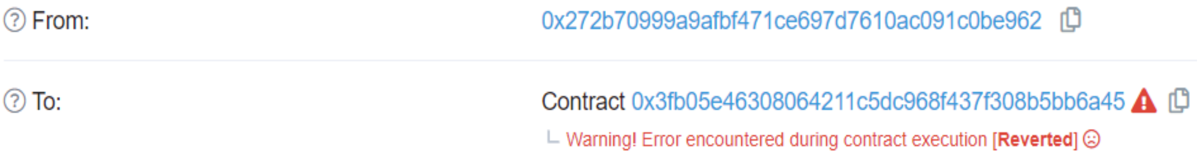
Figure 7: Details of a permissions granting transaction

This transaction has the following ID, which can be used to check it in the Ropsten Etherscan:

- *"0x48241d896e88b69c38fbe1ae4e3dcdf9f9206496d13715c27c73c8e57f97eacf".*

Publishing new records about business process models now is accessible not only to the owner but also to the "0x1E8d4Bdd6209Ebfd51aBbfcA12d9050E04A27adD" user account.

When a user, who does not have permission, tries to publish a new business process model to the collection, e.g. with "0x272B70999A9aFbF471Ce697D7610AC091C0be962" account, the following result is obtained (see Figure 8).



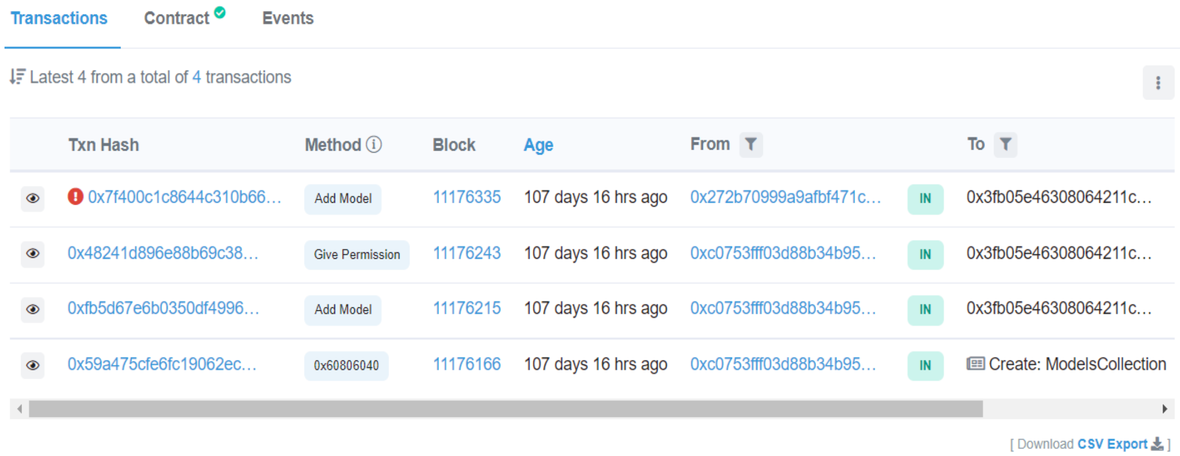
From:	0x272b70999a9afbf471ce697d7610ac091c0be962
To:	Contract 0x3fb05e46308064211c5dc968f437f308b5bb6a45 ⚠ └ Warning! Error encountered during contract execution [**Reverted**] ☹

Figure 8: Example of the unpermitted model publishing

This transaction has the following ID, which can be used to check it in the Ropsten Etherscan:

- *"0x7f400c1c8644c310b66119ca01c3daf71cb6eae4370c8ba46a40783e3752103e".*

All of the considered transactions can be seen on the smart contract page (see Fig. 9) [31].



Latest 4 from a total of 4 transactions

Txn Hash	Method	Block	Age	From		To
0x7f400c1c8644c310b66…	Add Model	11176335	107 days 16 hrs ago	0x272b70999a9afbf471c…	IN	0x3fb05e46308064211c…
0x48241d896e88b69c38…	Give Permission	11176243	107 days 16 hrs ago	0xc0753fff03d88b34b95…	IN	0x3fb05e46308064211c…
0xfb5d67e6b0350df4996…	Add Model	11176215	107 days 16 hrs ago	0xc0753fff03d88b34b95…	IN	0x3fb05e46308064211c…
0x59a475cfe6fc19062ec…	0x60806040	11176166	107 days 16 hrs ago	0xc0753fff03d88b34b95…	IN	Create: ModelsCollection

Figure 9: The list of smart contract transactions

Fig. 9 above demonstrates all of the successful and failed transactions of the smart contract [31].

4. Conclusion and Future Work

In this study, we proposed the approach to the organization of a decentralized blockchain-based business process model repository that could be used to provide a secure software solution to store and access the business process model collection in a tamper-resistant manner.

A review of the state-of-the-art has shown the complexity of the problem of managing large collections of business process models since the security and integrity of these enterprise knowledge assets are vital characteristics of the repository software intended to manage large business process model collections. There were covered multiple existing solutions in the domain of blockchain-based business process management, including decentralized systems for business process execution and cross-organizational collaboration. However, inter-organizational blockchain-backed solutions, intended to keep and share business process knowledge, have not been elaborated yet. As the result of blockchain platforms overview considering their smart contracts programming capabilities, was chosen the Ethereum platform was the pioneer and leader in the field of decentralized applications. Finally, there was proposed a conceptual model of a business process model repository based on blockchain technology.

The proposed approach is based on several formal definitions given to elaborate the domain of blockchain-based business process model collections. There were given definitions of the collection of business process models, the business process model record, the permissions mapping, and the collection management algorithms. These algorithms are used to add a new record about a business process model and configure access permissions. Formal definitions and algorithms were implemented as the smart contract prototype deployed to the Ethereum test network Ropsten. Also, there was created the prototype of a web application to access the smart contract.

Future work includes the elaboration of exchanging and trading capabilities of the proposed solution. Also, we plan to consider the non-fungible token (NFT) standard for storing and exchanging business models over the inter-organizational blockchain in a standardized and unified manner.

5. References

[1] Yan, Z., Dijkman, R., Grefen, P., Business process model repositories–Framework and survey. Information and software technology 54(4) (2012) 380–395.

[2] Elias, M., Design of business process model repositories: requirements, semantic annotation model and relationship meta-model. Stockholm University, 2015.

[3] About the Business Process Model and Notation Specification Version 2.0.2. URL: https://www.omg.org/spec/BPMN/2.0.2/.

[4] La Rosa, M. et al., APROMORE: An advanced process model repository. Expert Systems with Applications 38(6) (2011) 7029–7040.

[5] Yan, Z., Grefen, P., A framework for business process model repositories. In International Conference on Business Process Management. Springer, Heidelberg, 2010, pp. 559–570.

[6] Sakr, S., Awad, A., A framework for querying graph-based business process models. In Proceedings of the 19th international conference on World wide web, 2010, pp. 1297–1300.

[7] Dijkman, R., La Rosa, M., Reijers, H., Managing large collections of business process models-current techniques and challenges. Computers in Industry 63(2) (2012) 91–97.

[8] Dijkman, R. et al., Similarity of business process models: Metrics and evaluation. Information Systems 36(2) (2011) 498–516.

[9] Becker, M., Laue, R., A comparative survey of business process similarity measures. Computers in Industry 63(2) (2012) 148–167.

[10] López-Pintado, O. et al.: Caterpillar: A Blockchain-Based Business Process Management System. In BPM (Demos), 2017.

[11] López-Pintado, O., García-Bañuelos, L., Dumas, M., Business process execution on blockchain. In CEUR Workshop Proceedings 2144 (2018) 10–18.

[12] López-Pintado, O. et al., Caterpillar: A business process execution engine on the Ethereum blockchain. Software: Practice and Experience, 49(7) (2019) 1162–1193.

[13] López-Pintado, O. et al., Interpreted execution of business process models on blockchain. In 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference, 2019, pp. 206–215.

[14] Di Ciccio et al., Blockchain support for collaborative business processes. Informatik Spektrum 42(3) (2019) 182–190.

[15] Mühlberger, R., Extracting event logs for process mining from data stored on the blockchain. In International Conference on Business Process Management. Springer, Cham, 2019, pp. 690–703.

[16] Härer, F., Decentralized business process modeling and instance tracking secured by a blockchain. 26th European Conference on Information Systems. Portsmouth, UK, 2018.

[17] Fill, H. G., Härer, F., Knowledge blockchains: Applying blockchain technologies to enterprise modeling. In Proceedings of the 51st Hawaii International Conference on System Sciences, 2018, pp. 1–10.

[18] Hull, R., Blockchain: distributed event-based processing in a data-centric world. In Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, 2017, pp. 2–4.

[19] Viriyasitavat, W., Hoonsopon, D., Blockchain characteristics and consensus in modern business processes. Journal of Industrial Information Integration 13 (2019) 32–39.

[20] De Sousa, V. A., Corentin, B., Towards an integrated methodology for the development of blockchain-based solutions supporting cross-organizational processes. In 2019 13th International Conference on Research Challenges in Information Science, 2019, pp. 1–6.

[21] Milani, F., Garcia-Banuelos, L., Blockchain and principles of business process re-engineering for process innovation. arXiv preprint1806.03054, 2018.

[22] Macdonald, M., Liu-Thorrold, L., Julien, R., The blockchain: a comparison of platforms and their uses beyond bitcoin. COMS4507-Adv. Computer and Network Security, 2017.

[23] Fundamentals of Blockchains. URL: https://www.researchgate.net/publication/351116382_Fundamentals_of_Blockchains.

[24] Zheng, Z., An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems 105 (2020) 475–491.

[25] Oliva, G. A., Hassan, A. E., Jiang, Z. M. J., An exploratory study of smart contracts in the Ethereum blockchain platform. Empirical Software Engineering (2020) 1–41.

[26] The Best Smart Contract Platforms. URL: https://academy.shrimpy.io/post/the-best-smart-contract-platforms.

[27] Storing Documents on the Blockchain. URL: https://coincentral.com/storing-documents-on-the-blockchain-why-how-and-where/.

[28] Ethereum Networks. URL: https://ethereum.org/en/developers/docs/networks/.

[29] GitHub repository. URL: https://github.com/andriikopp/blockchain-repository.

[30] BPMN for research. URL: https://github.com/camunda/bpmn-for-research.

[31] Smart contract address. URL: https://ropsten.etherscan.io/address/0x3fb05e46308064211c5dc968f437f308b5bb6a45.

[32] Goods dispatch business process model. URL: https://raw.githubusercontent.com/freebpmnquality/bpmn_structuredness/main/input/dispatch.bpmn.