Towards the Tokenization of Business Process Models using the Blockchain Technology and Smart Contracts

Andrii Kopp and Dmytro Orlovskyi

National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61001, Ukraine

Abstract
Business process modeling helps organizations to capture their workflows visually as diagrams that could be then used to share best practices, identify inefficiencies in ongoing activities, instruct employees, use them as reference solutions, etc. Design and analysis of business process models are essential technologies of the Business Process Management approach, which is successfully adopted and practiced nowadays by many large and medium enterprises. Therefore, business process models should be considered as organizational assets that depict usable and competitive business solutions, which value could be proven by benchmarking. Single reference business process models or even collections of business process models are already accessible on Internet, sometimes for free or, usually, for purchasing because of the value of transferred knowledge. However, peer-to-peer exchange of business process models on a commercial basis is still far from a unification. At the same time, the tremendous growth of the cryptocurrency market and the adoption of Bitcoin by governments (first by El Salvador in June 2021) makes crypto-economics, also referred to as "tokenomics", usable for organizational knowledge sharing and exchanging without third party authorities, such as banks, or payment systems. Moreover, collaborating parties could reach a consensus when exchanging business process models using smart contracts and crypto-tokens to access shared knowledge artifacts. Therefore, this paper proposes an approach to business process model tokenization using blockchain technology and smart contracts. There was proposed as an Ethereum smart contract that combines features of the business process model collection and the non-fungible token. A prototype of a decentralized application was developed and its usage was demonstrated.

Keywords
Business Process Modeling, Blockchain, Smart Contract, Tokenization

1. Introduction: Related Work and Problem Statement

Nowadays digital transformation is a trend in enterprise management. In the first place, digital transformation is associated with Business Process Management (BPM) and its applications in business process modeling, and automation using BPM suites. However, business process modeling is used not only to draw executable workflows – they are usually simplified enough and describe mostly routine document flows. The main goal of business process modeling includes a visual representation of business activities as graphical diagrams to identify and understand ongoing workflows, find bottlenecks for improvement, and ensure communication between IT (Information Technology) staff and business stakeholders. Widely used reference models of typical enterprise business processes are used by organizations to adopt and tune concerning industry and internal needs. At the highest levels of BPM, maturity organizations have in their possession large collections of business process models that are extremely valuable for them and their competitors. Therefore, successful and time-proven business scenarios captured as process models could be sold to other organizations willing to achieve

CMIS-2022: The Fifth International Workshop on Computer Modeling and Intelligent Systems, May 12, 2022, Zaporizhzhia, Ukraine
EMAIL: kopp93@gmail.com (A. Kopp); orlovskyi.dm@gmail.com (D. Orlovskyi)
ORCID: 0000-0002-3189-5623 (A. Kopp); 0000-0002-8261-2988 (D. Orlovskyi)

positive BPM outcomes. Secure and peer-to-peer exchange of business process models could be organized using blockchain technologies, including cryptocurrencies and smart contracts. The only problem of business process models tokenization, i.e. representation of them as digital tradable assets on a certain crypto-platform [1], must be solved to reach such BPM-driven tokenomics. The object of this research includes sharing and exchange of business process models. The subject of research is the approach to business process model tokenization using blockchain and smart contracts.

This paper is structured as follows: in the introductory section is given an overview of related work (sub-section 1.1) and the problem statement is made (sub-section 1.2); section 2 outlines the approach to business process models tokenization based on smart contracts usage; section 3 contains results and their discussion regarding decentralized application prototype development and validation; conclusion and further research plans are outlined in section 4.

1.1. Related Work

1.1.1. Business Process Modeling

In general business processes are considered as structured collections of manual or automatic (by IT systems, e.g. BPMS, CRM, ERP, and others) executed activities necessary to achieve business goals and satisfy end customers [2]. According to the BPM approach, which focuses on the automation of business processes and support of human interaction with IT applications, there are different phases of business process analysis, modeling, implementation, and deployment to the execution environment of a certain IT system, monitoring, and evaluation [2]. Despite business process models could be defined in two ways: textual or visual [2], graphical diagrams are much more informative in describing how to process activities are triggered by events, which data objects are processed, which organizational units are responsible for process execution, and which outputs are produced [3]. Business process modeling is considered the approach to the depiction of current or future organization activities driven by events and control flow logic [4]. Also in [4], business process models are named as the key tools for process-aware information systems design, business process re-engineering, and service-oriented architectures design [4]. Business process models are also considered graphical knowledge resources and could be used as guidelines to introduce best practices for BPM adoption across multiple enterprises, as it is done by industry reference models that share knowledge of other organizations [4].

An extensive classification of business process modeling perspectives was given by J. Krogstie in [5]. There are object, communication, role, topological, functional, and behavioral perspectives. Some of the most well-known and widely used modeling standards, notations, and languages were covered in this classification: UML (Unified Modeling Language), IDEF0 (Integrated Definition for Functional Modeling), DFD (Data Flow Diagrams), EPC (Event-driven Process Chains), and BPMN (Business Process Modeling and Notation) [5]. A combined behavioral and functional approach was chosen as the most suitable for business process modeling, whereas the BPMN process diagramming notation has been adopted as the standard in BPM in general and, in particular, in business process modeling [5].

Even though EPC notation is still in use in the area of business process modeling, many users have replaced it with BPMN since it is a standard [6], moreover, most modern EPC modeling software tools support BPMN as the second or even alternative modeling notation (e.g. ARIS Express) [6]. As for other languages and standards, such as UML, DFD, and IDEF0, they did not become popular and are rarely used for business-oriented process modeling in practice [6]. In the last decade, BPMN has become a leading business process modeling notation [6], there are over 70 BPMN modeling tools listed on the “BPMN Tool Matrix” [7] and over 50 open-source tools related to BPMN listed on “Source Forge” [8]. BPMN is a complicated notation, there are four core symbols in use [9]:

- Activities. Tasks or units of work that have a certain duration.
- Events. Things that happen instantaneously and indicate when process instances begin (start events), complete (end events), or when something happens inside a process (intermediate events).
- Gateways. Model parallel (AND), exclusive (XOR), and inclusive (OR) branches of a process using combinations of splits and joins.
- Sequence flows. Model logical relations between elements, when one is followed by another.

Pools and lanes are used to model process resources: pools define business parties, i.e. boundaries of a business process, while lanes define roles, i.e. organizational units or persons, that take part in a business process execution [9]. There are also data objects and data stores (containers of data objects that could be databases for electronic objects or some places for physical objects) that could be depicted in BPMN models to represent information or material flows between activities [9]. As for limitations of the BPMN notation, authors of [9] do not recommend using OR-gateways without strong necessity because of their confusing logic, as well as to use data objects and data stores that make diagrams less readable and only useful when communicating to the IT development team for process automation.

Sample BPMN model of a goods purchase business process, which outlines considered symbols, is demonstrated in Fig. 1 below.

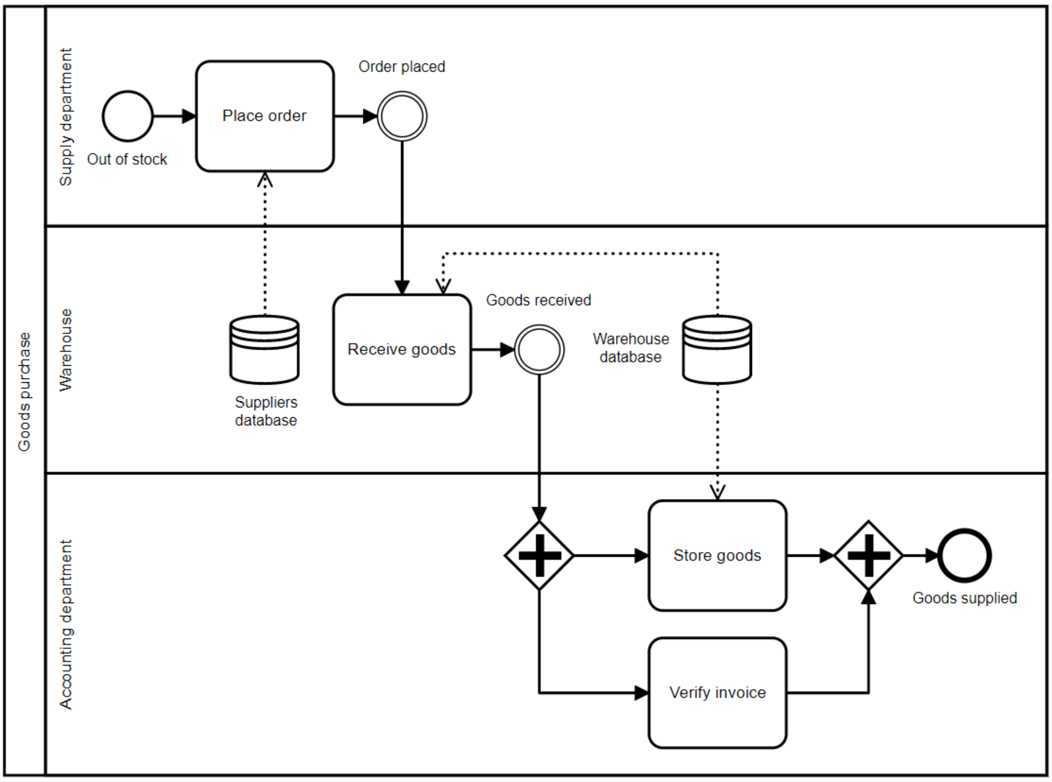

Figure 1: The BPMN model of a goods purchase business process

This diagram contains one pool “Goods purchase” that defined business process boundaries, three lanes “Supply department”, “Warehouse”, and “Accounting department” that define responsible roles for the process execution. As for core BPMN symbols, there are: start event “Out of stock” that starts process instance, intermediate events (e.g. “Goods received” and “Order placed”), end event “Goods supplied” that completes process instances, tasks (e.g. “Place order”, “Receive goods”, “Store goods”, and “Verify invoice”), split and join AND-gateways to implement parallel execution of “Store goods” and “Verify invoice” tasks (see Fig. 1).

1.1.2. Blockchain Technology

The term “blockchain” means an immutable or read-only data structure – a linked list of blocks, a directed acyclic graph (DAG), or a tree-like data structure, in which new data can be only appended at the end of the blockchain [10]. Blockchain systems are considered trustless distributed networks where a ledger is replicated over several nodes, each of which can participate in the decision-making process (i.e. a consensus protocol used to achieve an agreement between nodes when adding new blocks to the blockchain) in a decentralized and democratic manner [10]. Moreover, blocks contain transactions with certain data about transferred cryptocurrency, assets, tokens, etc. [10].

There are the following principles of blockchain technology that make it adaptable in many industries where security, transparency, and consistency are necessary features: decentralization, peer-to-peer communication, transparency, pseudonymity, irreversibility, and computational logic [11]. In general, blockchain is useful only when several entities are collaborating and sharing data since local copies are maintained on participating nodes to ensure consistency and tamper resistance [11].

The underlying idea of blockchain is using cryptography to link data blocks of referred distributed and decentralized chains. The first block is called the “genesis block”, it does not refer to any previous block in the chain [12]. However, each block contains data (usually as strings), nonce (the unique number usually related to mining – none generated repeatedly until meets consensus requirements of new block adding), and the hash value of all fields of the previous block (i.e. the cryptographic link to a preceding block, also referred as “previous hash”), the hash value of all fields of the current block [12]. The original consensus algorithm Proof-of-Work (PoW) was used to confirm transactions and add new blocks to the Bitcoin blockchain. This algorithm assumes miners (persons who share their computational power to support the network) are competing to confirm transactions and get rewarded in cryptocurrency [11]. Today PoW is used by such most popular cryptocurrencies like Bitcoin and Ethereum. However, Ethereum is planned to be switched to another popular consensus algorithm referred to as Proof-of-Stake (PoS). This algorithm overcomes high energy use since transactions are confirmed by validators who stake coins instead of sharing their computing resources. The higher stake is, the higher the probability to add a new block with pending transactions and getting rewarded [11].

However, in comparison to Bitcoin and other cryptocurrency-oriented blockchain platforms (such as Dogecoin, Bitcoin Cash, Litecoin, etc.), Ethereum was designed as a “world computer” with the ability to host smart contracts – programs to be executed within a blockchain platform [13].

1.1.3. Smart Contracts

Bitcoin as the first cryptocurrency was proposed in 2008 by Satoshi Nakamoto, who is presumably an anonymous person or group of persons hiding by this pseudonym. The difference from traditional payment systems is that electronic currency could be transferred in a peer-to-peer manner without a central party that needs to check the records of ownership. Blockchain technology helps to prevent the double-spending problem and enforce the validity of transaction records [13]. One more crypto-asset “token” is almost the same as the cryptocurrency – a bearer instrument used to transfer value between parties over the blockchain network; whereas tokens are created by a single party to represent a certain value, cryptocurrency is generated by the network as the reward for miners or validators [13].

Token technology was introduced and standardized by Ethereum and its smart contracts, which code describes how each token should work [13]. However, Ethereum was planned as a globally distributed computing network that uses publically stored immutable programs – smart contracts also referred to as decentralized applications [14]. However, decentralized applications usually include a client-side created using markup, style sheets, and JavaScript (JS). Decentralized applications (or DApps) use the Web3 JS library to interact with the smart contracts deployed on the blockchain [12]. DApps are immutable (as any data stored on the blockchain) and perform exactly as they were developed, without the possibility of fraud, downtime, censorship, or interference [14].

1.1.4. Tokenization

In the context of blockchain, tokenization means a representation of real physical or electronic assets digitally on the blockchain [15]. There could be commodities, real estate, ownership rights for arts or other collectibles, currency, or any other kinds of assets. As advantages tokenization offers faster and cheaper transaction processing, flexibility, decentralization, security, and transparency [15], but there are disadvantages, such as regulatory and legality issues, as well as technical barriers caused by the use of DApps [15]. There are two types of crypto-tokens [16]:

- Fungible Tokens (FT), which value is identical among all tokens, and which are exchangeable to each other (i.e. one FT token could be replaced by another FT token similarly to digital cash).

- Non-fungible Tokens (NFT) are unique and not equal to each other in value (i.e. each NFT token is different from others and cannot be replaced by any of them).

Another explanation of FT and NFT given in [16] says that dollar bills are exchangeable to other dollar bills, so they are fungible, whereas baseball cards or other collectibles are unequal in their value and cannot be replaced with another one.

From the technological point of view, both FT and NFT are implemented as smart contracts, which are programmed in a specific way. Several smart contract standards for tokens were developed by the Ethereum community [17]. The most popular contract standards are [17]:

- ERC20 standard for fungible tokens, which is suitable for multiple use cases, such as payment tokens, loyalty coins, gift cards, etc. A great example of NT implemented as the ERC20 token on the Ethereum blockchain is Tether USD, a stable cryptocurrency (also referred to as the "stablecoin") that digitally represents USD (United States Dollar) [18].
- ERC721 standard for non-fungible tokens, which is suitable for documents, land titles, digital identities, real estate, collectibles, etc. A great example of NFT implemented as the ERC721 token on the Ethereum blockchain is CryptoKitties, an Ethereum-based game in which players buy, sell, and breed collectible digital cats [19].

Nowadays OpenZeppelin organization provides extensive references for the development of ERC20, ERC721, and other smart contract standards [20].

1.2. Problem Statement

The problem of blockchain-driven cross-organizational business process modeling, versioning, and executing was considered by Härer in [21] and Fill in [22], Hull in [23], Viriyasitavat and Hoonsopon in [24], De Sousa and Corentin in [25], Milani and Garcia-Banuelos in [26], and others. Thanks to the considered blockchain advantages, inter-organizational storage of tokenized business process models provides collaborative parties with proof of authorship, censorship resistance, timestamping, and immutability. Smart contracts and token standards provide decentralized financial (DeFi) capabilities, such as peer-to-peer exchange of enterprise knowledge presented as business process models without the need of any third-party authorities, i.e. banks or payment systems. Therefore, the problem of business process model tokenization remains relevant and respective information technologies should consider the latest trends in blockchain technology and digital economics. Since in Spring 2021 NFTs got a focus in global crypto attention, mostly for collectibles and digital art (the entire market exceeds 130 USD by Spring 2021) [27], it seems like a great opportunity to extend the use cases of NFTs with tokenization of enterprise models, in particular – business process models.

The NFT better suites process models that are unique and unequal in terms of their syntactic and semantic properties, which could be used to define the value of shared models. There could be used SEQUAL (Semiotic Quality) framework [5] for the evaluation of syntactic and semantic validity and completeness of BPMN business process models given as graphic diagrams (i.e. images). Hence, the following business process model tokenization workflow could be used (see Fig. 2).

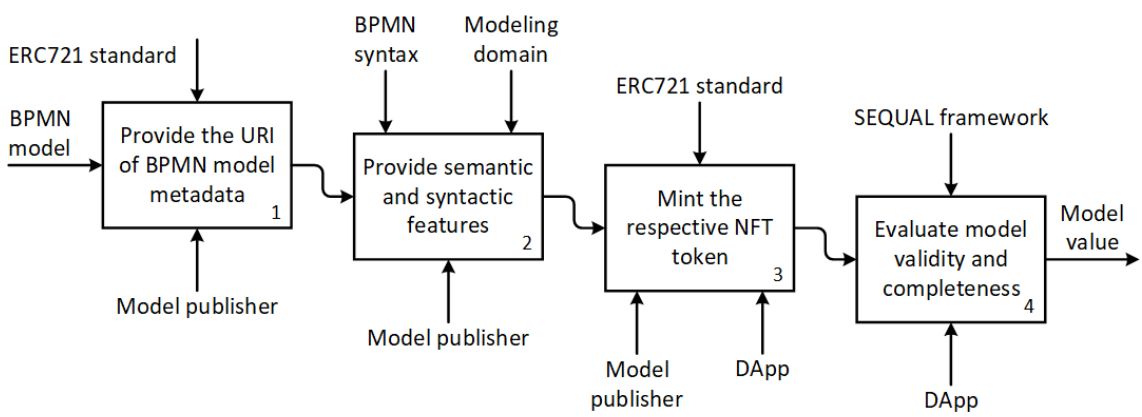

Figure 2: The business process model tokenization workflow

As for the blockchain platform, there could be chosen Ethereum as the pioneering and still leading smart contracts platform despite its competitors, such as Binance Smart Chain, Polkadot, Solana, and others [28]. Thus, the ERC721 standard may be used for business process model tokenization. As it is shown in Fig. 2, the URI (Uniform Resource Identifier) of a business process model should be used as the token URI. Besides the token URI, the ERC721 standard should be extended with syntactic and semantic features of business process models. After NFT is minted (i.e. published on the blockchain), syntactic and semantic features are used to evaluate the validity and completeness of a process model to define its value for collaborating parties (Fig. 2).

2. Tokenization of Business Process Models using Smart Contracts

2.1. Evaluation of Tokenized Business Process Models

According to [5], an extensive contribution to the domain of quality of business process models by J. Krogstie, syntactic and semantic qualities of business process models could be defined as following (according to the proposed specialization of SEQUAL framework for business process models):

- Syntactic quality is formulated as the correspondence of all statements (i.e. symbols, such as activities, etc.) in a business process model to the vocabulary and syntax of modeling notation – in our case BPMN.
- Semantic quality, in its turn, is defined as the correspondence between all statements and the modeling domain (i.e. a real business process).

In terms of formalisms proposed in [5], there could be presented phenomena of syntactic invalidity or syntactic incompleteness:

- Syntactic invalidity – when symbols are not part of BPMN notation (i.e. elements differ from events, activities, gateways, etc.).
- Syntactic incompleteness – when symbols do not obey BPMN syntax (i.e. elements connected improperly).

Formally, syntactic quality is defined as following [5]:

$$\text{Syntactic quality} = 1 - \frac{\# M \setminus L + M_{\text{missing}}}{\# M}, \tag{1}$$

where:

- $\# M \setminus L$ is the number of business process model M statements that do not correspond to the BPMN language L;
- M_{missing} is the number of missing statements that make a model syntactically incomplete (i.e. missing sequence flows or events);
- $\# M$ is the total number of business process model statements.

As for semantic properties, there are also validity and completeness phenomena [5]:

- Semantic invalidity – when statements are not part of the modeling domain.
- Semantic incompleteness – when statements are not correct or relevant to the modeling domain.

In [5] there are two metrics of semantic quality:

$$\text{Semantic validity} = 1 - \frac{\# M \setminus D}{\# M}, \tag{2}$$

$$\text{Semantic completeness} = 1 - \frac{\# D \setminus M}{\# D}, \tag{3}$$

where:

- $\# M \setminus D$ is the number of business process model M statements that do not belong to the modeling domain D;
- $\# D \setminus M$ is the number of model M statements that are not correct or relevant to the modeling domain D;
- $\# D$ is the number of statements in the modeling domain.

Therefore, a quality-oriented smart contract that may be used to assess tokenized business process models should consider formalisms of SEQUAL framework (1), (2), and (3) regarding syntactic and semantic quality of business process models given in [5]. Thus, each tokenized BPMN model should be described by the following tuples:

$$\left\langle N_{\text{invalid}}^{\text{synt}}, N_{\text{incomplete}}^{\text{synt}} \right\rangle, \quad (4)$$

$$\left\langle N_{\text{invalid}}^{\text{sem}}, N_{\text{incomplete}}^{\text{sem}} \right\rangle, \quad (5)$$

where:

- $N_{\text{invalid}}^{\text{synt}}$ is the number of syntactically invalid statements in a business process model;
- $N_{\text{incomplete}}^{\text{synt}}$ is the number of syntactically incomplete statements in a business process model;
- $N_{\text{invalid}}^{\text{sem}}$ is the number of semantically invalid statements in a business process model;
- $N_{\text{incomplete}}^{\text{sem}}$ is the number of semantically incomplete statements in a business process model.

In addition to (4) and (5), there should be noted the total number of statements N, which could be considered as equal to $\#M$ and $\#D$, since the syntax and semantics of real business process models are completing each other in order to reflect business activities.

Finally, there are following mapping should be noted:

$$Syntactic: tokenId \to \left\langle N_{\text{invalid}}^{\text{synt}}, N_{\text{incomplete}}^{\text{synt}} \right\rangle, \left(N_{\text{invalid}}^{\text{synt}} + N_{\text{incomplete}}^{\text{synt}}\right) \le N, \quad (6)$$

$$Semantic: tokenId \to \left\langle N_{\text{invalid}}^{\text{sem}}, N_{\text{incomplete}}^{\text{sem}} \right\rangle, \left(N_{\text{invalid}}^{\text{sem}} + N_{\text{incomplete}}^{\text{sem}}\right) \le N, \quad (7)$$

$$Total: tokenId \to N, \quad (8)$$

where $tokenId$ is the unique identifier of each tokenized business process model.

Whereas (6), (7), and (8) define the structure of business process model properties, there are following metrics should be calculated then:

$$\text{Syntactic validity} = 1 - \frac{N_{\text{invalid}}^{\text{synt}}}{N}, \quad (9)$$

$$\text{Syntactic completeness} = 1 - \frac{N_{\text{incomplete}}^{\text{synt}}}{N}, \quad (10)$$

$$\text{Semantic validity} = 1 - \frac{N_{\text{invalid}}^{\text{sem}}}{N}, \quad (11)$$

$$\text{Semantic completeness} = 1 - \frac{N_{\text{incomplete}}^{\text{sem}}}{N}. \quad (12)$$

Interpretation of calculated metrics (9) – (12) may be done using the Harrington scale [29] to transform crisp values of syntactic and semantic validity and completeness into linguistic values. To visualize obtained linguistic values, it is proposed to use “Green”, “Yellow”, and “Red” color codes for “Good”, “Satisfied”, and “Bad” values respectively (see Table 1).

Table 1

Translation of syntactic and semantic quality metrics into linguistic values and color codes

Linguistic value	Threshold	Color code
Good	Syntactic validity ≥ 0.8, Syntactic completeness ≥ 0.8, Semantic validity ≥ 0.8, Semantic completeness ≥ 0.8	Green
Satisfied	Syntactic validity ≥ 0.63, Syntactic completeness ≥ 0.63, Semantic validity ≥ 0.63, Semantic completeness ≥ 0.63	Yellow
Bad	Syntactic validity < 0.63, Syntactic completeness < 0.63, Semantic validity < 0.63, Semantic completeness < 0.63	Red

Obtained evaluation results will help collaborating parties define values and formulate prices of shared business process models for further exchange.

2.2. NFT-Compatible Smart Contract to Store Business Process Models

As it was outlined before, the ERC721 NFT standard [30] will be used for business process model tokenization. According to the problem statement, the ERC721 standard should be extended with the following behavior: model publishing and contacts data publishing (so parties can reach each other for collaboration). Before a model is published to the blockchain, i.e. the respective NFT is minted, special metadata JSON (JavaScript Object Notation) file should be published to the Internet and be accessible by a certain URI (later will be used as the token URI). Such JSON document accessible by the token URI might include the following properties [30]:

- Name (i.e. a brief description of a depicted business process).
- Description (i.e. a more detailed description of a depicted business process).
- Image URI (i.e. an image of a BPMN diagram).

Obviously token metadata stored in JSON files will not be stored on the blockchain. However, such an approach is considered the most efficient from the perspective of transactions' speed and cost [31]. Whereas to save tamper resistance provided by the blockchain technology, only the hash value of an entire document could be found using the secure algorithm (e.g. SHA-256 or others) and stored on the blockchain to verify the identity of the original document [31].

Then, to summarize, we may define two more mappings in addition to formalisms (6), (7), and (8) denoted earlier:

$$Party : address \to contacts, contacts \subseteq \{\Sigma_{UTF8}, \varnothing\}, \tag{13}$$

$$Hash : tokenId \to sha256(tokenURI), \tag{14}$$

where:

- $address$ is the user address in the Ethereum network;
- $contacts$ is the user's contact information outside the blockchain (e.g. email address, phone or messenger number, etc.) represented as the string value of UTF-8 characters set Σ_{UTF8} or even the empty characters set $\varnothing$ (i.e. the empty string) if a party decided to keep pseudonymity;
- $sha256$ is the hashing algorithm SHA-256 applied to the JSON document $tokenURI$.

Moreover, besides the (6) – (8) and (13) – (14) mappings, the smart contract should store a number of models $modelsCount$ used both as the capacity and the identifier of next minted token. In general, the minting or model publishing sequence diagram may look as following (see Fig.3).

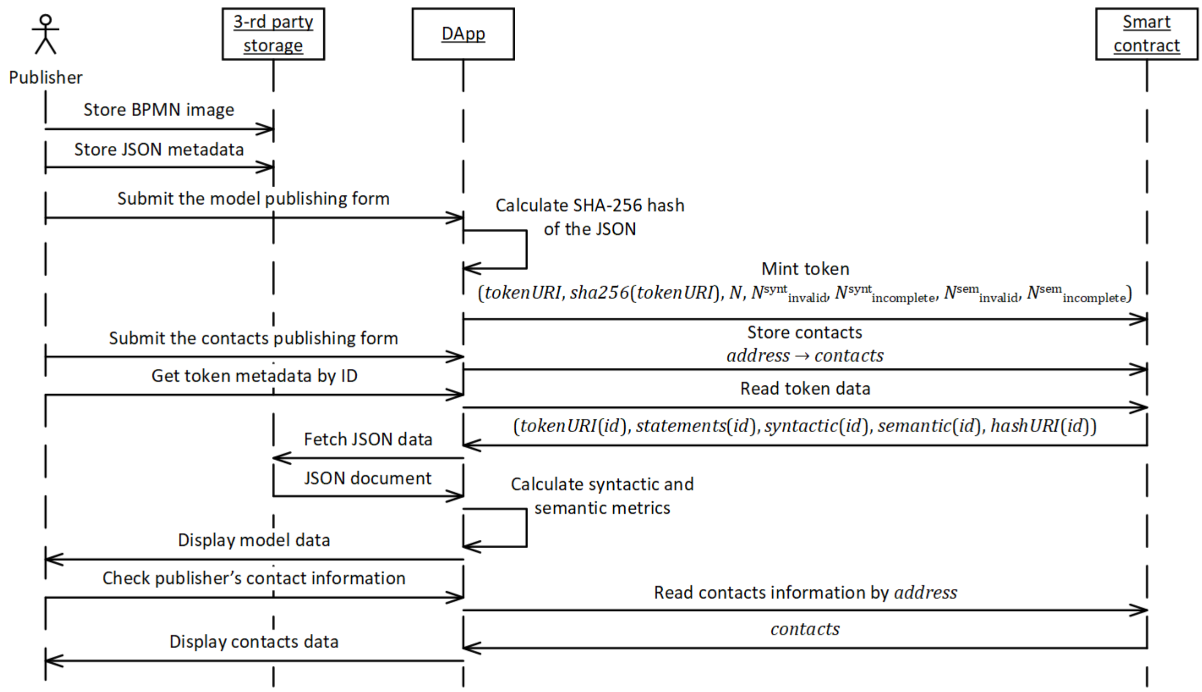

Figure 3: The model publishing sequence diagram

Thus, as depicted in Fig. 3 behavior should be implemented in the form of an ERC721-compatible smart contract that allows for the publication and read tokenized business process models and related data. In terms of UML class diagrams, the smart contract can be depicted as follows (see Fig. 4).

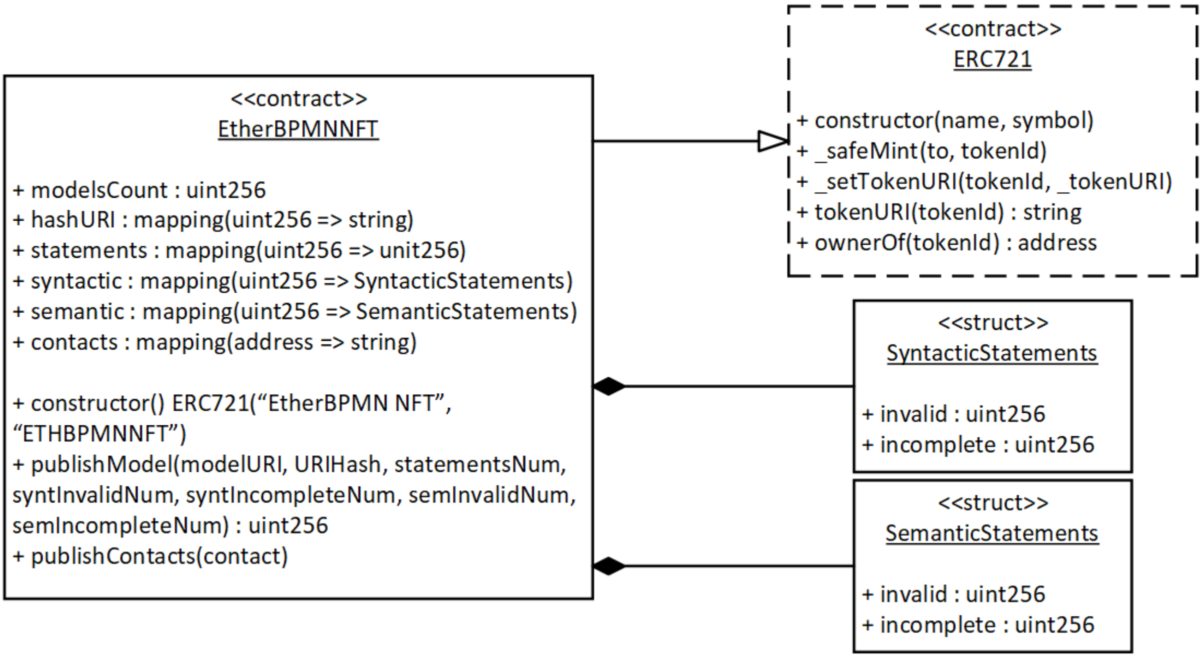

Figure 4: The class diagram of the smart contract

Contract "ERC721" with the dashed border line represents OpenZeppelin ERC721 implementation provided at [30] (see Fig. 4), which is used as the generic for developed contract "EtherBPMNNFT". The smart contract also contains two structures "SyntacticStatements" and "SemanticStatements" that serve as tuples of semantic and syntactic characteristics of tokenized BPMN models. As shown in Fig. 4 mappings of "EtherBPMNNFT" smart contract implement formalisms (6) – (8) and (13) – (14), while functions "publishModel" and "publishContacts" are used to mint NFT tokens and store collaborator contact information respectively. Some of ERC721 standard functions can be called from the developed contract, these functions are also mentioned in Fig. 4.

3. Results and Discussion

3.1. Design and Development of a Decentralized Application Prototype

Considered smart contract is barely useless without having a special decentralized application that allows ordinary users to interact with the blockchain. The following use cases should be supported by the DApp prototype for BPMN model tokenization (see Fig. 5).

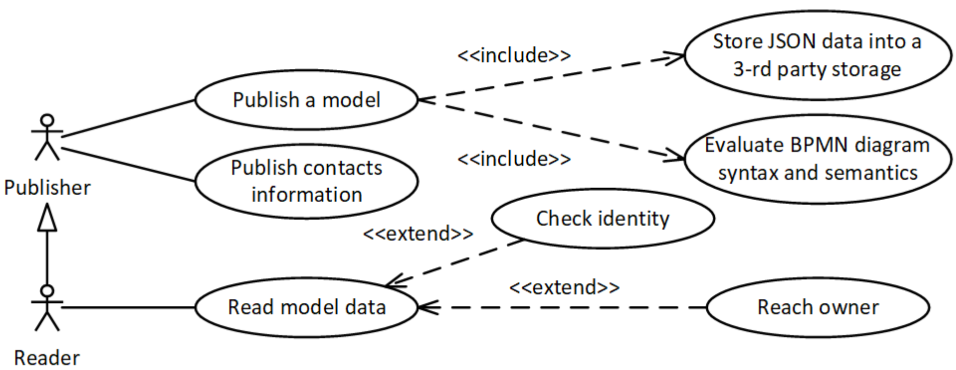

Figure 5: The use cases of DApp prototype

The system architecture of the decentralized application is similar to any client-server web application, whereas instead of an application server and persistent storage the smart contract and blockchain ledger is used respectively [12]. The structure of the DApp prototype is shown in Fig. 6.

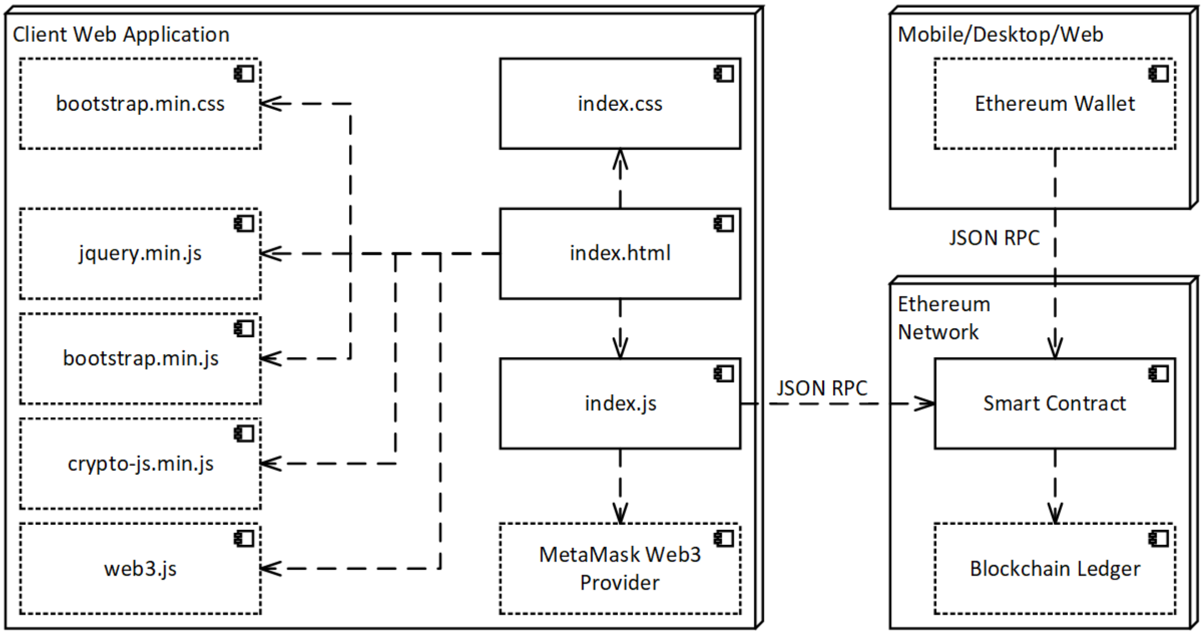

Figure 6: The system architecture of DApp prototype

Components with dashed border lines are third-party libraries, such as Bootstrap for user interface, CryptoJS for SHA-256 calculation, Web3 for interaction with the Ethereum platform, Ethereum blockchain ledger itself, MetaMask Web3 provider (as the Google Chrome extension), and also some Ethereum mobile, desktop, or web wallet that can be used to hold business process models in the form of ERC721-compatible NFTs, which are tradable and exchangeable as any other digital crypto-assets.

3.2. Validation of a Decentralized Application Prototype

Developed prototype of a decentralized application for business process model tokenization that can be used only with MetaMask or another Web3 provider. In the authors' opinion, MetaMask is the easiest and the most popular tool to work with Ethereum networks, either the main network or test network, since it only requires the installation of the Google Chrome extension. According to the sequence diagram (Fig. 3), the tokenization procedure starts with the storing of the JSON metadata and BPMN diagram image in the third-party non-blockchain storage. Let us imagine we are intending to store BPMN shown in Fig. 1. It is image is already given and could be stored somewhere on the Internet with the permanent URI. As for other metadata properties, the following could be given:

- Name: "Goods purchase".
- Description: "The BPMN model of a goods purchase business process".

Therefore, respective JSON (as well as the BPMN diagram image) could be stored in the GitHub repository of this project and may look like the following (see Fig. 7).


```
{
    "name": "Goods purchase",
    "description": "The BPMN model of a goods purchase business process",
    "image": "https://raw.githubusercontent.com/andriikopp/blockchain-repository/main/nft/models/0.png"
}
```


Figure 7: The JSON document representing the NFT metadata

The URI of future NFT is following – "https://raw.githubusercontent.com/andriikopp/blockchain-repository/main/nft/models/0.json". Now, after the non-blockchain data is stored, let us fill and submit the model publishing form in the DApp (see Fig. 8).

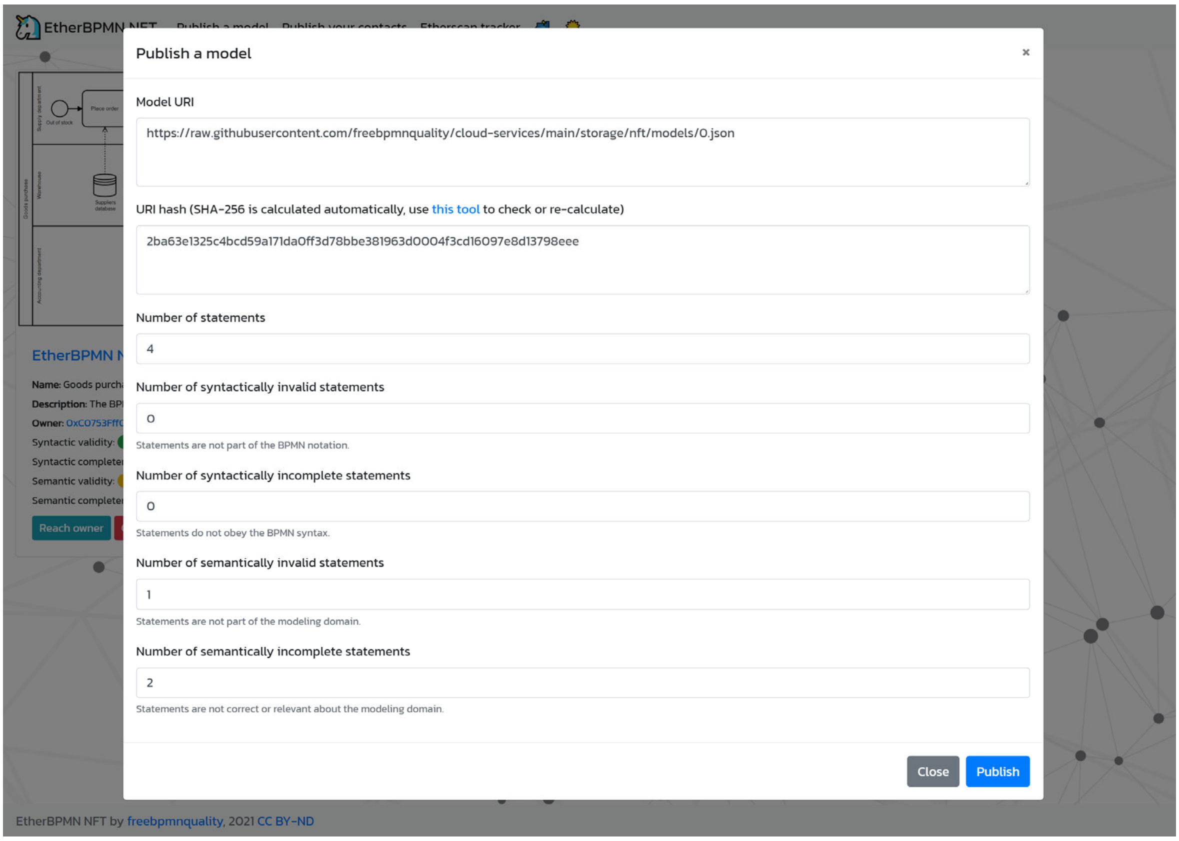

Figure 8: The model publishing form of the DApp

When the “Publish” button is pressed, the MetaMask appears with the request to sign a transaction as it is shown in Fig. 9. In case of the transaction is successfully mined, a respective token will be minted and can be accessed by the address “0xabc33640b17def441cfb455efa1c3f8f490f4616” and ID “0” (since it is the first NFT in a collection). A respective token could be added to MetaMask for future exchange with other parties (see Fig. 9).

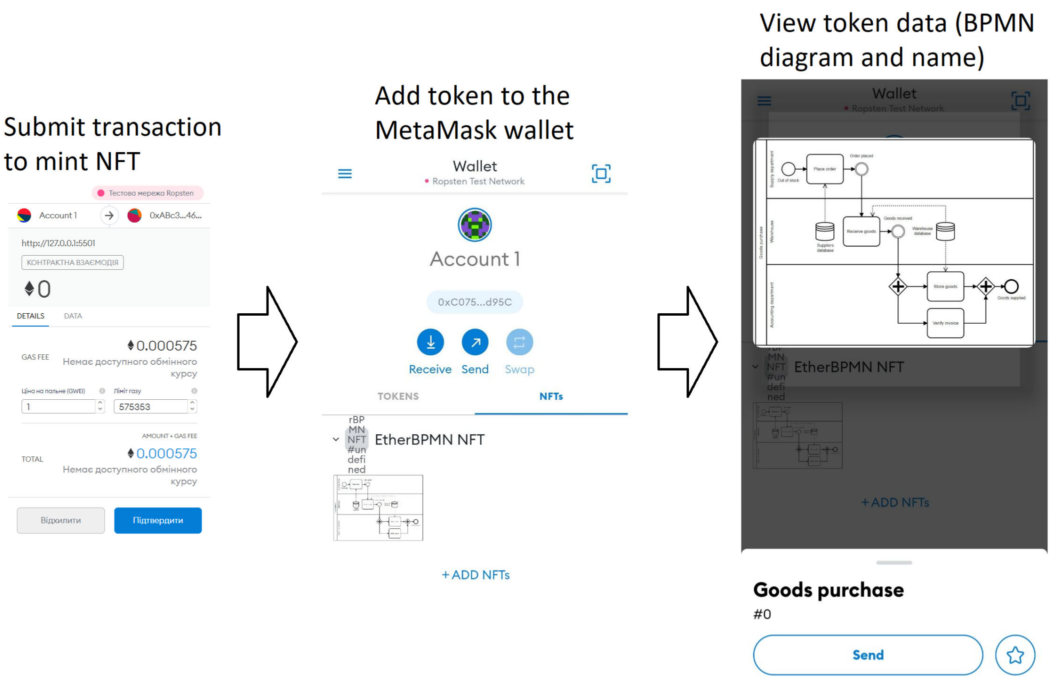

Figure 9: Tokenized business process model in the MetaMask wallet

Now BPMN diagram shown in Fig. 1 can be kept in the MetaMask wallet or any other Ethereum wallet that supports ERC721 tokens, exchanged with other Ethereum users, or even traded using NFT exchanges and marketplaces like other crypto-tokens that represent collectibles or digital art.

The developed DApp prototype also displays tokenized business process models with all given JSON-based data and owner's address, but also with the specific characteristics of BPMN diagrams: quality metrics of syntactic and semantic validity and completeness, a hash value used to ensure the identity of business process model metadata, and owner's contact information (if it was preliminary published to the smart contract, of course) as it is shown in Fig. 10.

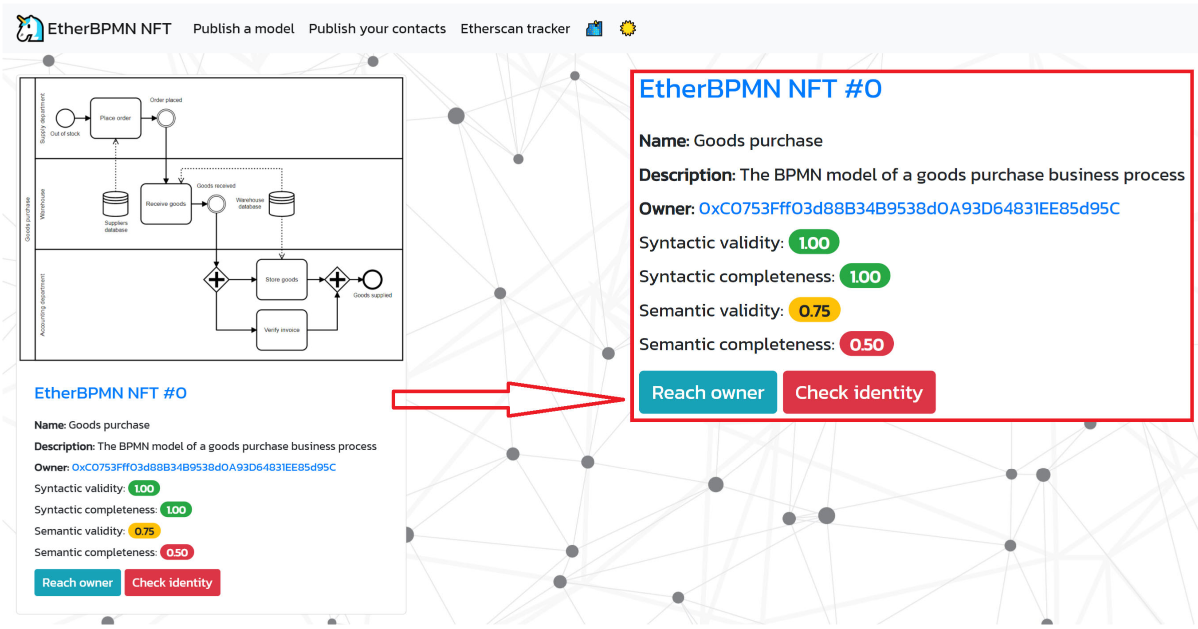

Figure 10: The homepage of developed DApp

According to Fig. 10, each tokenized BPMN model are displayed syntactic and semantic metrics that correspond to (9) – (12) formulas and color codes given in Table 1. Besides metrics, it is possible to reach the owner by retrieving its contact information (in case it was provided) and checking identity by comparing the SHA-256 hash value (stored on the blockchain in the smart contract mapping) to the actual SHA-256 hash value of JSON requested by token URI.

The form of contacts information publishing, as well as examples of requested contact information and SHA-256 hash of the NFT data, are shown in Fig. 11.

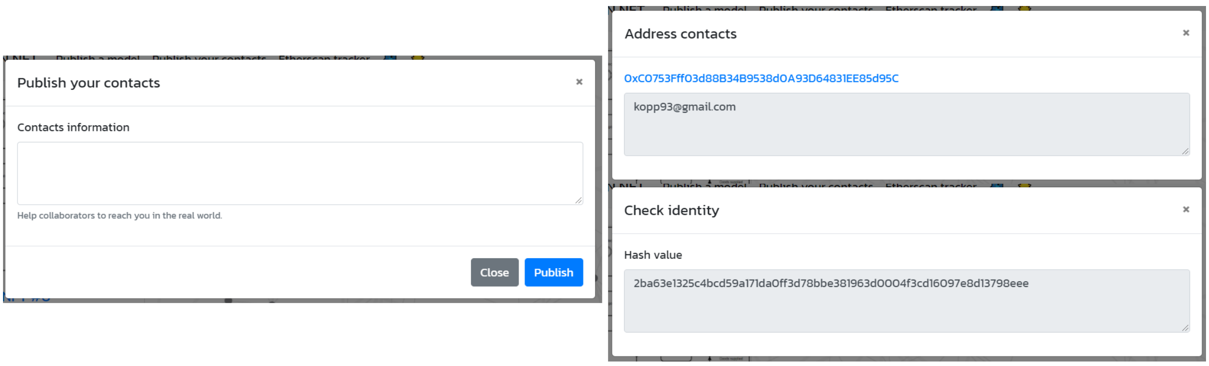

Figure 11: The contacts publishing form, contacts request window, and identity check window

Provided quality metrics may help to define the value of the business process model when exchanged, while the hash value may help to check whether the BPMN has been tampered with by the previous owner. Currently, the DApp is at the software prototype stage being under construction and testing [32]. The MetaMask or another wallet with an in-built Web3 provider should be installed to use the DApp, as well as the Ropsten testnet account should be present for basic usage.

4. Conclusion and Future Work

In this research paper, we proposed the approach to business process model tokenization based on blockchain technology and smart contracts. Based on the performed state-of-the-art overview, the BPMN business process modeling notation was chosen to describe tokenized business process models as the most widely used and considered standard in the BPM industry. Essentials of blockchain technology were considered to prove the relevance of business process model tokenization, and the essentials of smart contracts and decentralized applications were overviewed as well. Two standards of tokens – ERC20 and ERC721, which represent fungible and non-fungible tokens respectively were considered to select the appropriate token standard for BPMN diagrams. Based on features of NFTs, the ERC721 standard has been chosen, as well as Ethereum as the pioneering and still leading smart contracting platform has been chosen for implementation.

To provide tokenized business process models with specific features useful to define their value for exchanging and trading, essential syntactic and semantic quality metrics were used: validity and completeness [5]. Hence, originally provided by OpenZeppelin ERC721 smart contract for NFTs has been extended to keep syntactic and semantic properties of business process models, SHA-256 hash values of token metadata documents to ensure identity, and owner contact information to ensure collaboration of parties.

Developed DApp prototype allows to publish a model as the NFT, review already published NFTs, including model names, descriptions, BPMN diagram images (preview and full size), quality metrics, owner addresses, request owner contact information (if provided), check the identity of tokenized BPMN models, and share own contact information if necessary. The smart contract has been deployed to the Ropsten Ethereum test network, while the DApp is under development and testing.

Future work in this area includes the development of the decentralized marketplace and exchange for BPMN models as NFTs, as well as a more rigorous evaluation of tokenized business process models using special methods and algorithms, rather than human judgment. The DApp should evolve into the full-scale ecosystem of tokenized BPMN diagrams that could be traded and swapped in the same way it could be done with cryptocurrency and NFTs nowadays.

5. References

[1] B. Hines, What Types of Assets May be Tokenized?, in: Digital Finance: Security Tokens and Unlocking the Real Potential of Blockchain, John Wiley & Sons, 2020, pp. 47–60.

[2] C. Gerth, Business Process Modeling, in: Business Process Models: Change Management, Lecture Notes in Computer Science, Springer, 2013, pp. 14–23. doi:10.1007/978-3-642-38604-6.

[3] A.-W. Scheer, Strategic Business Process Analysis, in: ARIS – Business Process Modeling, Springer Science & Business Media, 2013, pp. 7–20. doi:10.1007/978-3-642-57108-4.

[4] A. Bitkowska, The relationship between Business Process Management and Knowledge Management – selected aspects from a study of companies in Poland, in: R. Gabryelczyk, T. Hernaus (Eds.), Business Process Management: Current Applications and the Challenges of Adoption, Cognitione Foundation for Dissemination of Knowledge and Science, 2020, pp 169 – 193. doi:10.7341/20201616.

[5] J. Krogstie, Quality of Business Process Models, in: Quality in Business Process Modeling, Springer, 2016, pp. 53–102. doi:10.1007/978-3-319-42512-2.

[6] T. Allweyer, BPMN – A Standard from Business Process Modeling, in: BPMN 2.0: Introduction to the Standard for Business Process Modeling, BoD – Books on Demand, 2016, pp. 9–14.

[7] BPMN Tool Matrix. URL: https://bpmnmatrix.github.io/.

[8] SourceForge – Search results for “bpmn”. URL: https://sourceforge.net/directory/?q=bpmn.

[9] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, Essential Process Modeling, in: Fundamentals of Business Process Management, Springer, 2018, pp. 75–116. doi:10.1007/978-3-662-56509-4.

[10] M. H. Rehmani, Introduction to Blockchain Systems, in: Blockchain Systems and Communication Networks: From Concepts to Implementation, Springer Nature, 2021, pp. 3–14. doi:10.1007/978-3-030-71788-9.

[11] J. Jayapriya, N. Jeyanthi, Distributed Consensus Protocols and Algorithms, in: K. Saini, P. R. Chelliah, D. K. Saini (Eds.), Essential Enterprise Blockchain Concepts and Applications, CRC Press, 2021, pp. 15–38.
[12] H. L. Gururaj et al., Blockahin: A New Era of Technology, in: G. Shrivastava, D.-N. Le, K. Sharma (Eds.), Cryptocurrencies and Blockchain Technology Applications, John Wiley & Sons, 2020, pp. 3–24.
[13] T. Laurence, Second generation applications of Blockchain technology, in: Introduction to Blockchain Technology, Van Haren, 2019, pp. 69–82.
[14] C. Blackburn, The Which and How's!, in: Ethereum Blockchain Revolution Explained: Understanding Ethereum Technology For Beginners, Christopher Blackburn, 2020, pp. 5–13.
[15] I. Bashir, Tokenization, in: Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Packt Publishing Ltd, 2020, pp. 567–614.
[16] B. Ramamurthy, Tokenization of assets, in: Blockchain in Action, Simon and Schuster, 2020, pp. 227–248.
[17] I. Roy, Building a Blockchain Wallet for Fungible and Non-Fungible Assets, in: Blockchain Development for Finance Projects: Building next-generation financial applications using Ethereum, Hyperledger Fabric, and Stellar, Packt Publishing Ltd, 2020, pp. 21–57.
[18] Tether USD (USDT) Token Tracker. URL: https://etherscan.io/token/0xdac17f958d2ee523a2206206994597c13d831ec7.
[19] Cryptopedia Staff, CryptoKitties: A Pioneer in Ethereum Gaming and NFTs, 2021. URL: https://www.gemini.com/cryptopedia/cryptokitties-nft-crypto-ethereum-token.
[20] Tokens – OpenZeppelin Docs. URL: https://docs.openzeppelin.com/contracts/2.x/tokens.
[21] F. Härer, Decentralized business process modeling and instance tracking secured by a blockchain, in: Proceedings of the 26th European Conference on Information Systems (ECIS2018), Portsmouth, UK, 2018, pp. 1–17. doi:10.5281/zenodo.2585717.
[22] H. G. Fill, F. Härer, Knowledge blockchains: Applying blockchain technologies to enterprise modeling, in: Proceedings of the 51st Hawaii International Conference on System Sciences, 2018, pp. 1–10. doi:10.24251/HICSS.2018.509.
[23] R. Hull, 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. doi:10.1145/3093742.3097982.
[24] W. Viriyasitavat, D. Hoonsopon, Blockchain characteristics and consensus in modern business processes, Journal of Industrial Information Integration 13 (2019) 32–39. doi:10.1016/j.jii.2018.07.004.
[25] V. A. De Sousa, B. Corentin, 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. doi:10.1109/RCIS.2019.8877045.
[26] F. Milani, L. Garcia-Banuelos, Blockchain and principles of business process re-engineering for process innovation, arXiv 1806.03054 (2018). doi:10.48550/arXiv.1806.03054.
[27] S. Jaitly, What are NFTs and Why are They Becoming Popular?, 2021. URL: https://medium.com/tribalscale/what-are-nfts-and-why-are-they-becoming-popular-c3ca2c84a4b3.
[28] The Shrimpy Team, The Best Smart Contract Platforms, 2021. URL: https://academy.shrimpy.io/post/the-best-smart-contract-platforms.
[29] A. Voronin (Ed.), Multicriteria Problems in Complex Systems Management, in: Multi-Criteria Decision Making for the Management of Complex Systems, IGI Global, 2017, pp. 17–44. doi:10.4018/978-1-5225-2509-7.ch002.
[30] ERC721. URL: https://docs.openzeppelin.com/contracts/2.x/erc721.
[31] B. Whittle, Storing Documents on the Blockchain: Why, How, and Where, 2018. URL: https://coincentral.com/storing-documents-on-the-blockchain-why-how-and-where/.
[32] The GitHub repository of a software prototype. URL: https://github.com/andriikopp/blockchain-repository/tree/main/nft.