VIII International conference
"Information Technology and Implementation" (IT&I-2021)
Kyiv, Ukraine

Towards an Approach to Organization of Decentralized Business Process Model Repository

Andrii Kopp, Dmytro Orlovskyi and Sergey Orekhov

Structure

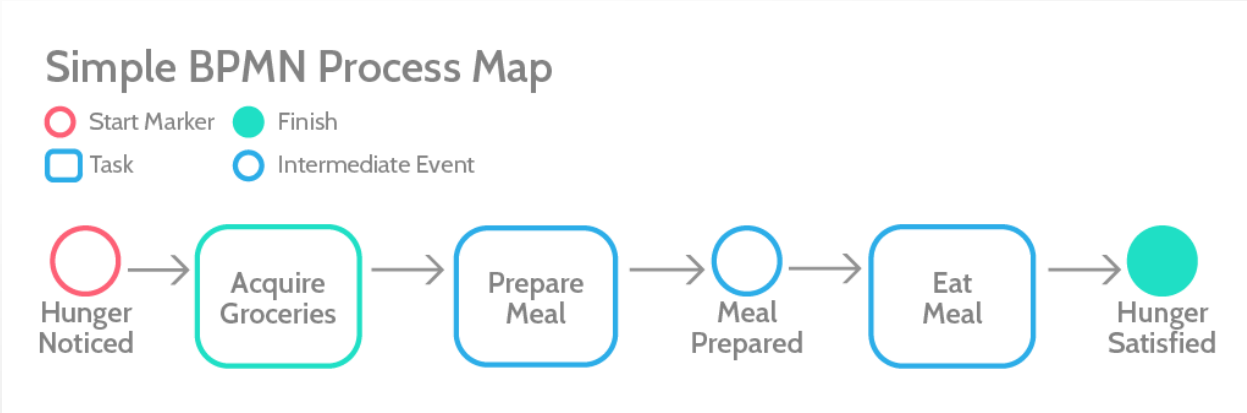
1. Motivation
    1. Business Process Model Repository
    2. Blockchain Technology
    3. Smart Contracts and Decentralized Applications
2. Problem Statement
3. Materials and Methods
    1. Collection of Business Process Models
    2. Business Process Model Record
    3. Permissions Mapping
    4. Managing a Collection of Business Process Models
    5. Algorithms of a Blockchain-based Business Process Model Collection
4. Results and Discussion
    1. Ethereum Smart Contract Prototype
    2. Web Application Prototype
    3. Validation of a Decentralized Application Prototype
5. Conclusion and Future Work

Motivation

- Large companies tend to keep extremely large collections of business process models.
- Availability, integrity, and security are vital for enterprise collections of business process models.
- Blockchain technology can be utilized to keep and manage business process models securely and stably.

Business Process Model Repository

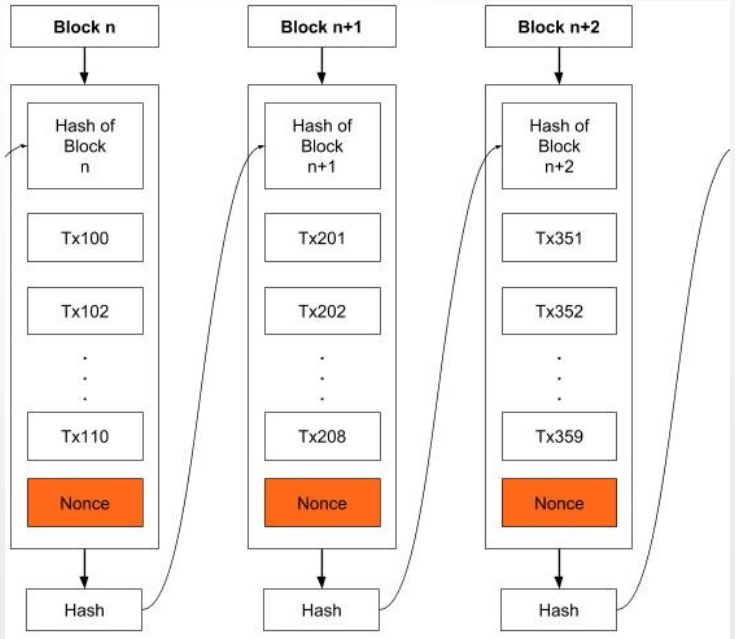
- A "business process model repository" is the software used to store, search, and manage business process models.
- Support of the BPMN – a de-facto standard business process modeling notation, displaying graphical and textual process descriptions.
- Access control and integrity fulfillment are critical for corporate assets.



https://www.process.st/bpmn-tutorial/

Blockchain Technology

- Stored data is consolidated into blocks that contain hash values of blocks generated before, which creates a chain of irreversible and immutable blocks.
- Data authenticity and consistency could be proven by checking the conformity of hash values back to the initial block of the whole blockchain.
- Blockchain transactions do not need a "trusted entity" for processing, they cannot be altered and could be easily traced.

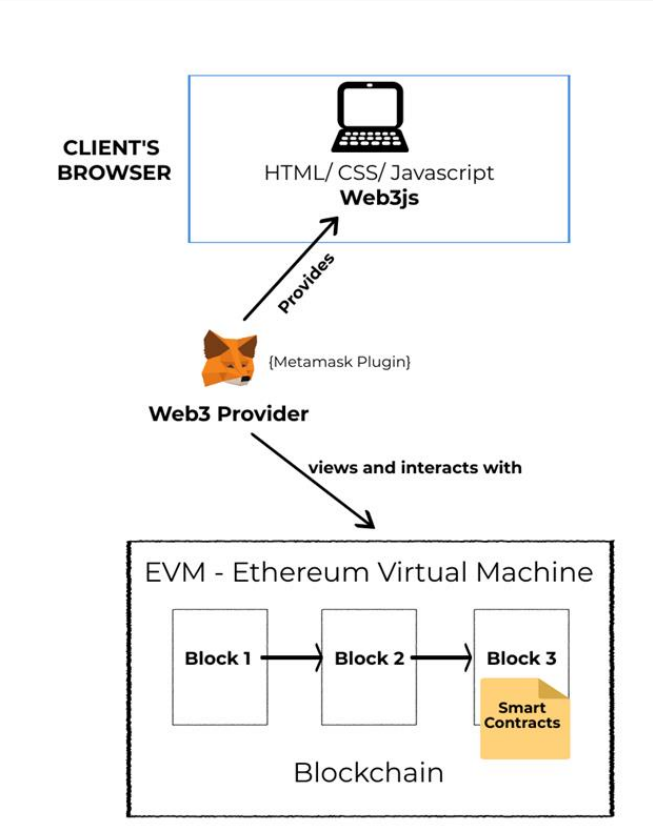


https://www.tutorialspoint.com/blockchain/index.htm

Smart Contracts and Decentralized Applications

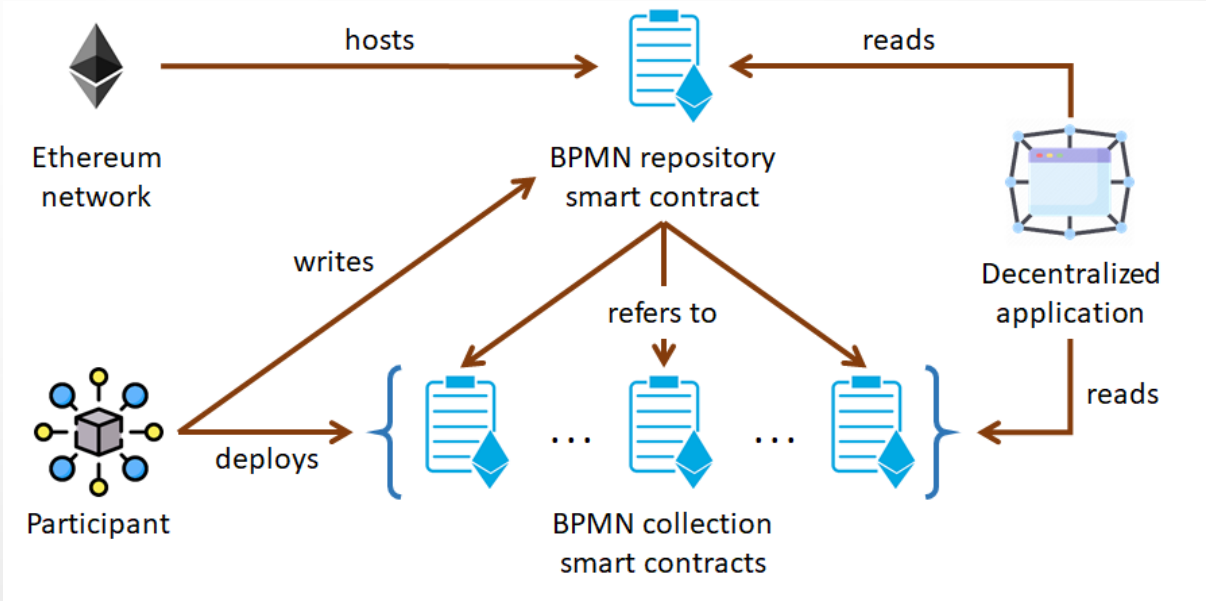
- "Smart contracts" are computer programs that run in the blockchain platform and record immutable transactions into the blockchain.
- Ethereum and other platforms that support smart contracts are called "programmable blockchains", which are used to create decentralized applications or "DApps".
- DApps use smart contracts as back-end code and blockchains as databases in contrast to traditional applications backed by centralized servers.

https://alastria-es.medium.com/comparison-of-dlt-platforms-be84950d339d



Problem Statement

- 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.
- A proof-of-concept smart contract and corresponding DApp should be implemented.



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:

$$C = <own, MR, m, P, A>,$$

- own – is the owner of a collection of business process models;
- MR – is the list of records about business process models;
- m – is the number of business process model records;
- P – is the permissions mapping, which defines users allowed to access this collection;
- A – is the set of algorithms used to manage a collection.

Business Process Model Record

- Business process model record is the structure that consists of business process model attributes:

$$MR_i = <t_i, u_i, h_i, desc_i, ind_i, ts_i>,$$

- t_i – is the title of a business process model;
- u_i – is the direct URL used to access a BPMN 2.0 document;
- h_i – is the hash value of a BPMN 2.0 content (SHA256 is used);
- $desc_i$ – is the annotation or brief description of a business process model;
- ind_i – is the industry name to which a business process belongs;
- ts_i – is the timestamp that shows when the record was made.

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 or not:

$$P\colon addr \rightarrow \{0, 1\},\ addr \in Addr,$$

- $addr$ – is the address in a blockchain network;
- values 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 $P(addr) = 1$.

Managing 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:

$$A = <add, set>,$$

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

Algorithms of a Blockchain-based Business Process Model Collection

Add a new business process model

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.

Set access permissions

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

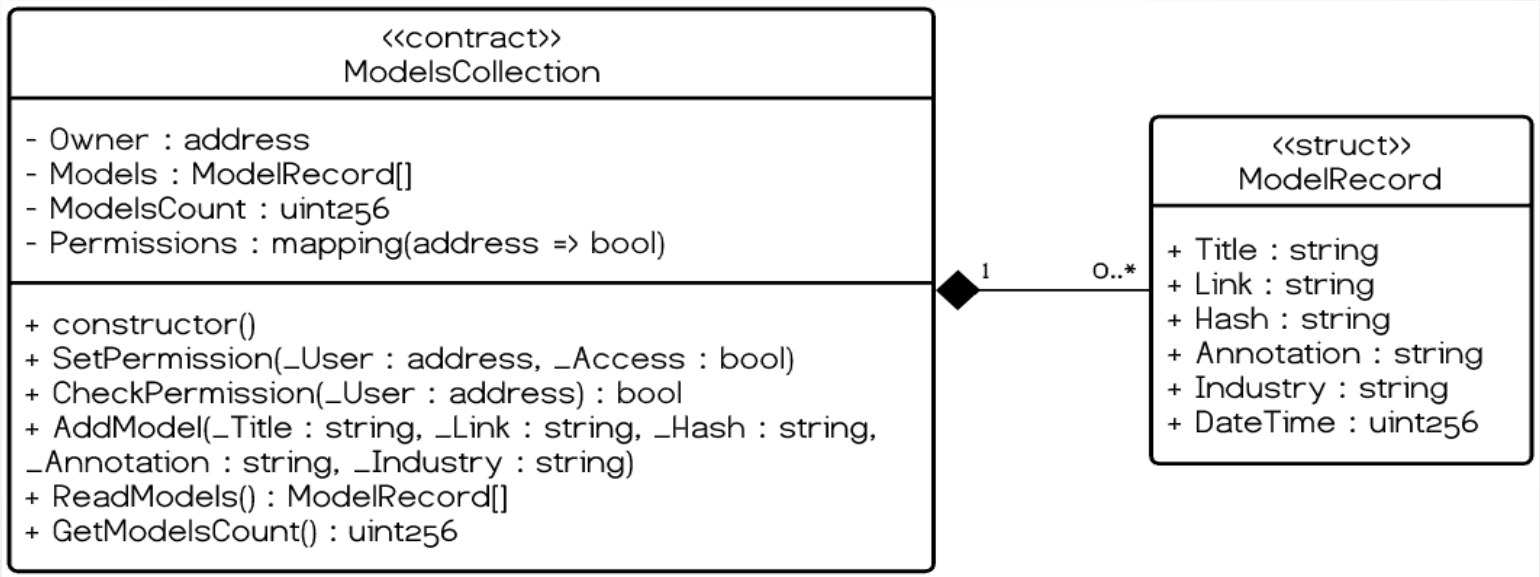
Step 2. Compare obtained sender's address $send \in Addr$ to the collection owner's address $own \in Addr$. If compared addresses are equal, then proceed to Step 3 and finish.

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, $P(user) = access$, $user \in Addr$.

Ethereum Smart Contract Prototype

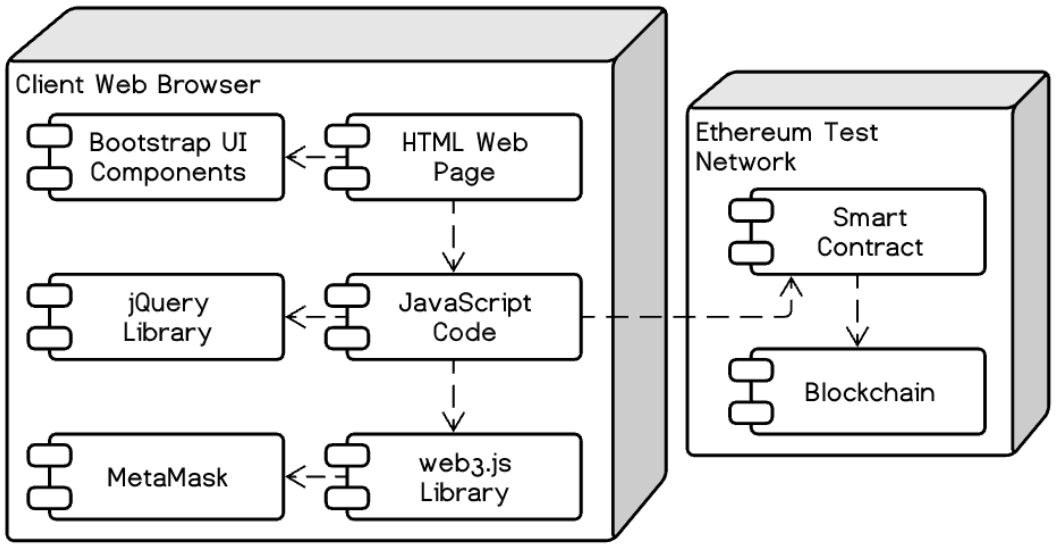
- Solidity
- Remix IDE
- Ropsten (Ethereum test network)
- MetaMask



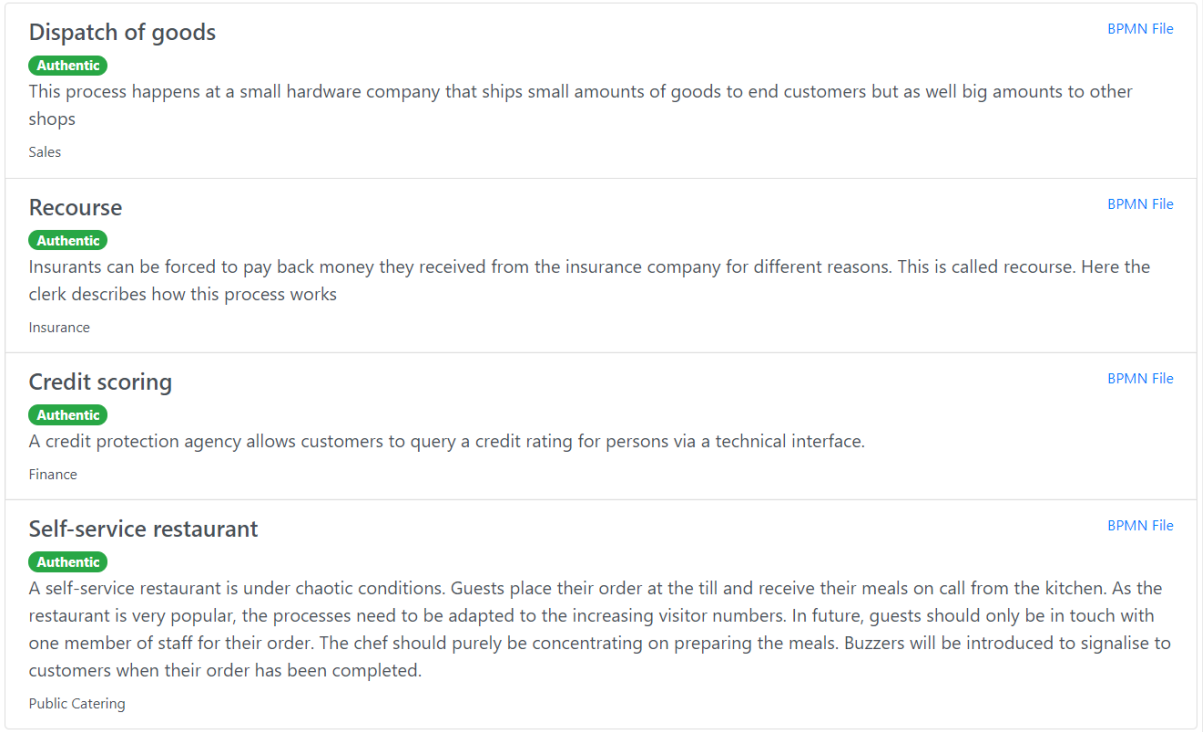
Structure of the smart contract prototype

Web Application Prototype

The system architecture of the web application prototype



Business process models requested from the smart contract

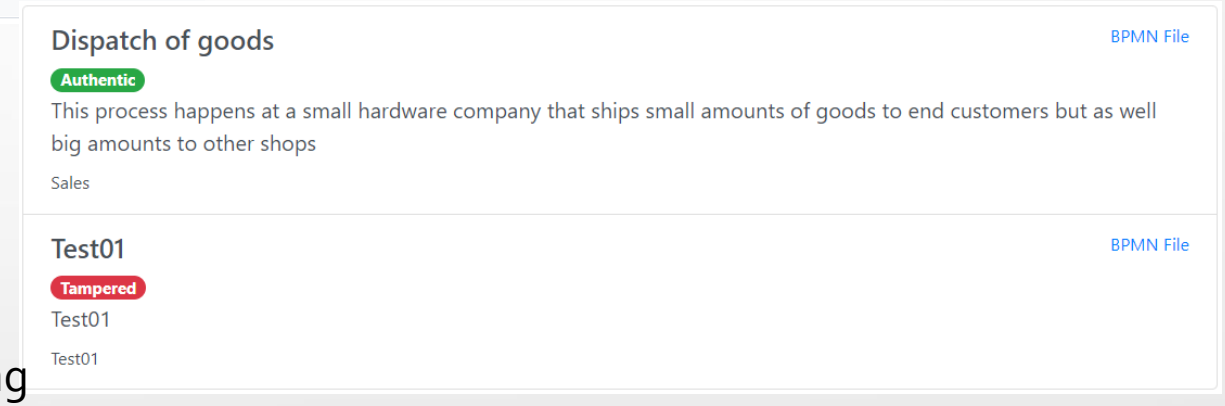


Validation of a Decentralized Application Prototype

Details of a business process model publishing transaction

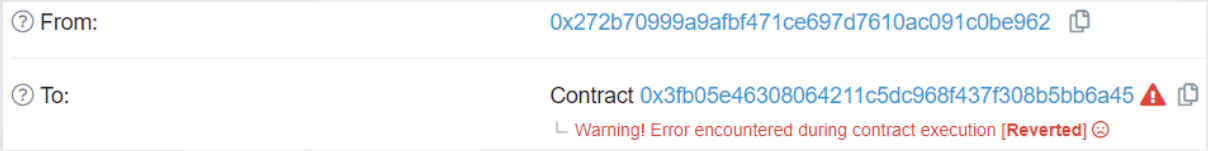
#	Name	Type	Data
0	_Title	string	Test01
1	_Link	string	https://raw.githubusercontent.com/freebpmnquality/bpmn_structuredness/main/input/dispatch.bpmn
2	_Hash	string	bd7ffce0d0b77dc74f8fab0b69c8b10d0956bdbd1901297d2c58bf7404d25400
3	_Annotation	string	Test01
4	_Industry	string	Test01

Example of tampered business process model detection



Smart contract deployed to the testnet:
https://ropsten.etherscan.io/address/0x3fb05e46308064211c5dc968f437f308b5bb6a45

Example of the unpermitted model publishing



Conclusion and Future Work

- We proposed formal definitions of the collection of business process models, the business process model record, the permissions mapping, and the collection management algorithms.
- Suggested algorithms allow to add a new record about a business process model and to configure access permissions.
- Formal definitions and algorithms were implemented as the smart contract prototype deployed to the Ethereum test network Ropsten.
- 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.

Thank you for your attention!