An Application of Blockchain Distributed Systems for Supply Chains in the Pharmaceutical Industry

Aleksa Mirković*, Marina Nenić*, Dušan Gajić*, Branko Terzić*, and Ivan Luković*
* University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
aleksa.mirkovic@uns.ac.rs, marina.nenic@uns.ac.rs, dusan.gajic@uns.ac.rs, branko.terzic@uns.ac.rs, ivan@uns.ac.rs

***Abstract*—This paper describes an application, built on top of the Hyperledger Fabric blockchain framework, for efficient and reliable support of the supply chain in the pharmaceutical industry. The presented application showcases how blockchain's immutable ledger, peer-to-peer (P2P) network, consensus algorithms, and smart contracts all combine and synergize to create a secure, reliable, auditable, and cheaper way of performing business. The application is developed on the Proof of Concept (PoC) level, using the Hyperledger Composer framework. It enables communication with a private Hyperledger Fabric network via Create, Read, Update, and Delete (CRUD) operations. The developed PoC shows that the use of the Hyperledger Composer framework allows simple creation of business networks which include participants, assets, transactions, and relationships between these entities. Our research shows that Hyperledger Composer significantly reduces the amount of time needed to develop a Hyperledger Fabric blockchain-based software solution. However, for more customized approaches, the Hyperledger Fabric platform needs to be used directly on the infrastructure level. When using Fabric, smart contracts are written on the lower level of abstraction, using the Go programming language, instead of the high-level JavaScript code used in Composer. Plans for future work include implementing the solution directly on the Hyperledger Fabric platform, as well as extending the available functionalities of the application to cover an even wider set of use cases.**

I. Introduction

Data in distributed systems are processed and stored on multiple computing and storage nodes, which can be spread across different locations in space. The main advantage of the distributed approach is that, if one of the nodes fails and becomes unavailable, data can still be accessed through other nodes which are participating in the distributed network. Further, computational and memory resources of all nodes in distributed systems may be combined to offer more robust and reliable solutions to larger problem instances. The distributed approach improves the stability and availability of computer systems, but it doesn't resolve on its own some of the typical problems or conflicts which occur in performing communication and transactions between multiple parties. For example, when different parties are communicating in distributed systems, every party has its own data and, thus, it is possible that the parties would happen to have different data regarding same transactions for whatever reason. Further, when it comes to performing financial transactions, banks are currently unavoidable intermediaries. They provide services that are often expensive and slow.

Blockchain is a distributed ledger which can efficiently record transactions between multiple parties in a verifiable and tamper-resistant way, thus solving the problem of trust in distributed systems. It was first introduced by Satoshi Nakamoto in 2008 [1]. The blockchain distributed ledger technology eliminates the need for intermediaries in transactions between untrusting parties, at the same time offering the aforementioned advantages of distributed databases in an append-only manner. Applying blockchain technologies in business systems provides greater transparency and safety of transactions and data in general. Besides that, blockchain allows the development of computer systems that are more fault tolerant and don't require intermediaries, thus leading to quicker and cheaper transactions [1]. In blockchain systems, every participant has its own copy of the data which he trusts and maintains. For every transaction that occurs, all relevant actors must validate the corresponding data and agree on it, in order for the transaction to be added to the blockchain.

The research presented in this paper uses the blockchain technology in order to improve the processes in the pharmaceutical industry supply chain – starting with the producers and finishing with the customers. Blockchain technology is selected as a basis for the presented approach due to its numerous advantages in the considered context. For example, speed of service and availability are of high importance when large number of transactions is happening between participants in the supply chain. Further, since the quality of operations in the pharmaceutical industry directly affects peoples' health, it is important to know where each of the products came from, who produced and managed it, and, finally, who delivered the product to the end customer.

In business applications, participants are not anonymous, which is in contrast to the use of public blockchains for cryptocurrencies, where anonymity is crucial [1]. Further, there are different roles for participants in the supply chain networks, e.g., small, medium, and large companies, shipping companies, and regulators. Therefore, in this context, there is a need for private and permissioned blockchain networks, in which every participant has specific roles and verified identity.

The main goal of the research presented in this paper is to create an appropriate model for a business network

which implements the aforementioned supply chain for the support of operations in the pharmaceutical industry. The presented results show how the use of blockchain in the considered context can significantly improve speed, cost, reliability, and robustness of service.

The paper is organized as follows. Fundamentals of the blockchain technology are described in Section II. In Section III, we describe the technological tools that were used to implement the solution presented in the paper. Implementation of the conceptual software solution is described in Section IV. The closing section of the paper offers main conclusions and offers some directions for further work.

II. THEORETICAL BACKGROUND

The blockchain distributed system was first introduced in [1]. Blockchain is a shared, distributed, and replicated ledger. This ledger can be also looked upon as a database. Unlike traditional databases, where every party who takes part in business dealings has its own data, blockchain enables all participants to share the same data, and provides the means of confirming that the data is authentic. The basic and most important elements of blockchain system are the distributed network, ledger, consensus algorithms, and smart contracts [1, 8, 9, 10].

A. The distributed network

Participants in a blockchain system communicate through a peer-to-peer network [1]. At least two different network types can be differentiated: public and private [5]. When it comes to public networks, each participant has the right to add entry in the database and read from it. In such networks, where privacy and anonymity is crucial, adding entries and achieving consensus goes mainly through a proof of work or a proof of stake algorithm [1, 2, 3]. On the other hand, private networks imply different access right for the network nodes. Private networks do not provide anonymity, which means that the identities of the participants are known. This property enables usage of wider range of consensus algorithms and quicker transaction confirmation. Differences between private and public networks are shown in Table 1.

B. Ledger

Every transaction in the system is registered in the ledger and it cannot be erased or changed without other participants being aware of this action. Before they are appended to the blockchain, transactions are organized into blocks. Apart from transactions, blocks in the blockchain also contain a block header. Block header of each block contains a reference to the previous block in the blockchain, as well as a root of the Merkle tree [6] where transactions are stored (see Fig. 1). Reference to the previous block is given as a hash pointer and it is calculated based on the content of the previous block's header.

TABLE I
COMPARISON OF PUBLIC AND PRIVATE BLOCKCHAIN NETWORKS

Network type	Public	Private
Access	Open access database for both writing and reading	Reading and writing possible with permission
Confirmation of transaction	Slower	Faster
Network validity maintence	Proof of work/ proof of stake	Many
Participants' identity	Anonymous	Familiar
Maintence	Expensive	Cheap
Number of nodes	Millions	Up to a few hundred
Example	Bitcoin	Hyperledger Fabric

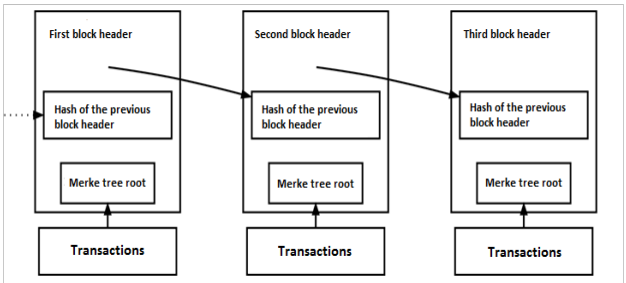

Figure 1. Structure of a block in a blockchain.

As previously mentioned, the chain of blocks is a structure resistant to change. Changes of the transactions in one block would result in different hash of that block. Consequently, all following block hashes in the chain would be changed and in that way the activity would be recognized as malicious and thus rejected by the rest of the network. Merkle tree is a structure which is used to store the data inside the blocks. In order to form a Merkle tree, each transaction is hashed and that is how the leaves on the first level of the tree are created. On the next level, hashes that are found in the leaves are combined in pairs and hashed again. This process is repeated until the root hash is created (see Fig. 2).

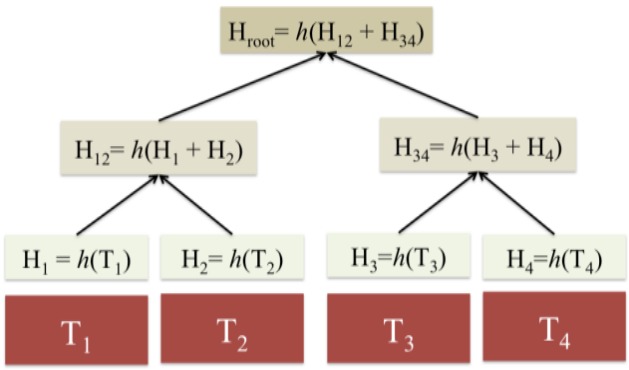

Figure 2. The Merkle tree.

C. The consensus algorithms

The process of keeping the ledger synchronized across the network, or, in other words, ensuring that the ledger is updated only when the transactions are approved by the proper participants is called consensus [7].

When it comes to traditional way of doing business, there are problems of trust among the participants and questioning of the data validity. These problems can be solved by the introduction of mediators, but this approach is often both expensive and time-consuming. In order to exclude mediators from the business process, blockchain imposes certain rules that need to be followed for the transaction to be added into the ledger (see Fig. 3). Proof of work [2], proof of stake [3] and Practical Byzantine Fault Tolerance (PBFT) [8] are some of the many algorithms which are used to reach consensus in block chain networks.

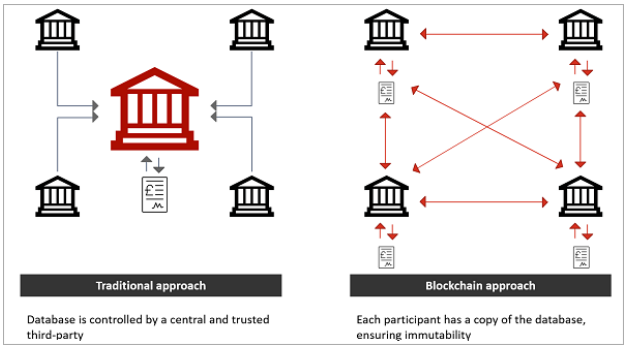

Figure 3. Traditional centralized approach of keeping records versus the blockchain distributed approach.

1) Proof of work

Proof of work is a consensus algorithm used to ensure proper functioning of public blockchain networks [1, 2]. It was introduced by Satoshi Nakamoto in [1]. The idea is that the node inside the network must invest process time (central processing unit (CPU) time) in order to send data for verification to the rest of the network. In order to add a block to the blockchain, the node which is sending data must solve complex computational problem, while, for the rest of the network, it is quite simple to check whether the offered solution to the problem is correct. The first node which solves the problem and offers the proof of work is called the winning miner and it appends the new block to the end of the chain.

2) Proof of stake

Proof of stake is an alternative algorithm to the proof of work, which does not require complex computations and resulting time and energy consumption [3]. Unlike in the proof of work system, the proof of stake does not impose competition between mining nodes. Each node, which wants to validate the block, has to invest a certain amount of the cryptocurrency used in the network. When choosing the validator, the network takes into consideration the amount of cryptocurrency that the candidate nodes possess.

3) Practical Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance [8] is one of the algorithms used for solving the Byzantine Generals Problem [9]. The preposition is that some of the nodes may be unavailable or send wrong data and that messages may not be arriving in the same order as they were sent. The algorithm requires that within the network there must be at least $3f+1$ nodes, where f is the maximal number of faulty nodes. Every node gets to decide for itself whether the message it received is correct or not. Then, in order for message to be recognized as valid, $2f+1$ nodes have to confirm its correctness.

D. Smart contracts

Smart contracts are basically computer programs used to support complex business cooperation between entities in blockchain networks. These contracts are business contracts translated in program code and thus automatically ensure fulfillment of contractual obligations. Smart contracts require from the blockchain system to offer a Turing-complete language for writing programs to be executed on the blockchain network [7].

III. TECHNOLOGY

Hyperledger [11] is chosen as the blockchain technology for the implementation of the constructed supply chain model for the following reasons.

Hyperledger Fabric [7] is a blockchain framework and one of the Hyperledger projects hosted by the Linux Foundation. Unlike most other blockchain implementations, which are public and with anonymous participants, Fabric is intended to be a private, permissioned network, where identities are known and transactions visibility can be limited to only those participants who are relevant for the given transaction. Further, Fabric provides the ability to create separate channels allowing participants to execute private transactions. This is an important business requirement which public blockchain networks, such as Bitcoin [1] or Ethereum [12], simply cannot provide. Since Fabric is intended as a foundation for developing applications or solutions with a modular architecture, it allows components, such as consensus algorithms and membership services, to be plug-and-play. Furthermore, Fabric leverages container technology to host smart contracts, called “chaincode”, that comprise the application logic of the system. Hyperledger Fabric allows users to create blockchain networks on a low level of abstraction, whereas Hyperledger Composer [4] allows easier development of Fabric applications. Composer also provides a simple way to model network participants and transactions. Its focus is on the efficient implementation of business logic. Composer uses its own modeling language for this purpose. Unlike Fabric, which uses Go for chaincode implementation, Composer allows users to write business logic in JavaScript. Composer also provides its own query language. Besides the blockchain part of the system, it also contains a NodeJS server which handles requests towards the blockchain and allows generation of Angular2 frontend web applications.

IV. IMPLEMENTATION

In order to achieve the main goals of the research presented in this paper, it is first necessary to identify all the relevant participants in the supply chain. An appropriate model for the identified participants, as well as assets, which are the subject of executing transactions in the chain, should then be developed. Thereafter, it is necessary to define permissions for participants, as well as their rights to manage assets. Finally, the implemented model should be used to configure and deploy a permissioned blockchain network that supports a predefined basic set of actions which are fundamental for the pharmaceutical supply chain. Main focus and questions posed in the research consider transactions which are used to represent the stages in the exchange of goods. These stages include sending orders, approving orders, preparing products, and shipping of products. In addition, creation or generation of a simple application which communicates with the deployed blockchain network is also required.

Several types of participants were identified while observing business dealings in pharmaceutical industry: end users, retail, wholesale and transport companies, and control factors (The Medicines and Medical Devices Agency and National Health Insurance Fund). In the presented solution, the participant in the network is modeled as an abstract type which is identified through email address and has account balance. All other groups of users inherit this abstract type (see Fig. 4).

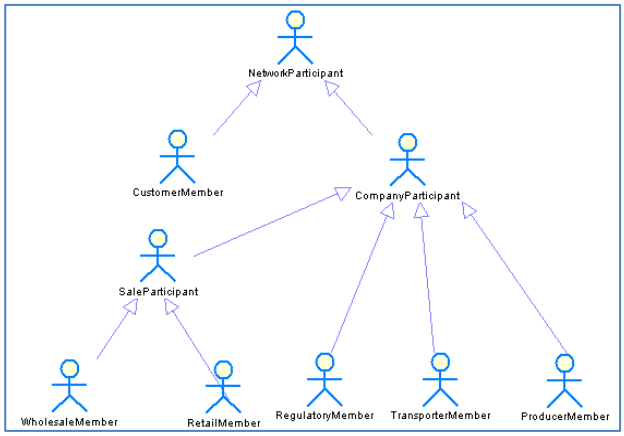

Figure 4. The network participants model.

Properties of participants representing retail or wholesale companies are expanded with features of the company. This group of users are modeled so that they have an array of incoming order forms so as to keep track of received orders. Besides that, they have a list of concrete products which are supposed to be in their possession (ordered products).

As far as the digital goods, i.e. assets, are concerned, product classes, product instances, and order with its items can be identified. Model of digital goods which was developed using the Hyperledger Composer is presented in Fig. 5.

```
asset Product identified by productName{
  o String productName
  o String productBarcode
  o String productDescription
  o ProductType productType
  o Double productPrice optional
}
asset ProductInstance identified by productInstanceId {
  -->Product product
  o String productInstanceId
  -->NetworkParticipant owner
  o DateTime producedDateTime optional
  o Double productInstancePrice
  o ProductStatus status optional
}
asset OrderItem identified by orderItemId{
  o String desc optional
  o String orderItemId
  -->Product product
  -->ProductInstance [] instances
  o Double amount
}
asset Order identified by orderId{
  o String orderId
  o Double totalPrice optional
  o DateTime createdTime optional
  o OrderStatus orderStatus default = "CREATED" optional
  --> OrderItem[] items
  -->NetworkParticipant sender
  -->NetworkParticipant receiver
}
```

Figure 5. The digital goods (assets) model.

Since it is important to know how each package of a product was transfered through the supply chain, it is necessary to lower the level of identification when compared to traditional business systems. Each product represents a class of concrete products and concrete product represents a single product itself. According to this, concrete product references the product class to which it belongs. Orders reference the sender and the receiver and also feature an array containing items that were ordered.

Business dealings in the pharmaceutical industry involve a lot of operations. The solution presented in this paper is focused on those operations which enable transfer of products among the participants.

SendOrder transaction implies sending of order form to the intended agent. If the order form has not been already sent and if there are enough resources, the order form is placed among receiver's incoming order forms.

ApproveAndPrepareOrder operation checks whether the receiver of the order form has the sufficient amount of each required product. If this verification passes successfully, then product instances which are to be delivered are added to the every order form item.

ShipOrder operation changes the status of product instances which will be sent and puts them in incominig product instances of the purchaser.

DeliverOrder is executed by special courier service or the employees of the company which sells the product. In this step, the purchaser account balance is decreased for the total price of the order form, while the same amount of money is added to the seller's account balance. Further, the owner of the product instances is changed.

The definition of the network also includes the access rules for the described assets and are defined as follows. Every agent of the network has insight into the product class and wholesale companies can have special prices for certain partners. Manufacturers or distributers can put the new product into circulation, whereas regulatory

bodies have the authority to introduce or withdraw the product. Every company can create the order form. Sender and receiver are the ones who have insight into the concrete order.

Based on the abovementioned models and access rules, which make up the Business network definition, simple frontend application is generated along with the REST (Representational State Transfer) API (Application Programming Interface). An example of a form for adding assets which is taken from the presented solution is shown in Fig. 6.

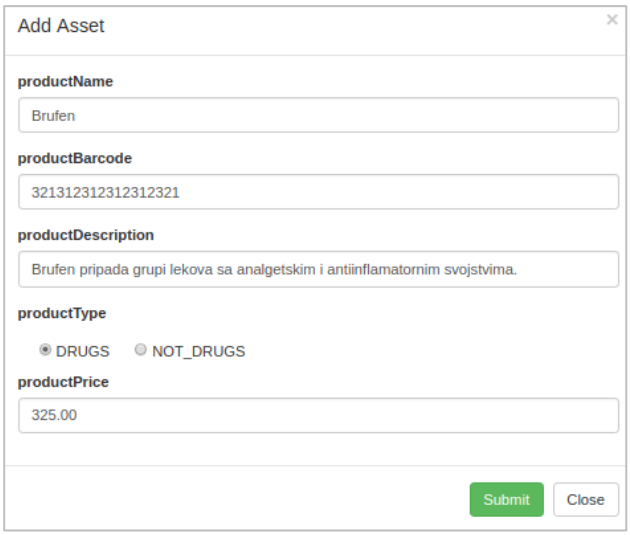

Figure 6. Form for the addition of new products.

V. CONCLUSIONS

In this paper we presented the fundamentals of the blockchain distributed systems and we proposed a solution which uses the advantages of private, permissioned blockchain systems in order to solve some of the problems in the pharmaceutical supply chain. We discussed many actors involved in the distribution of medical products and the complex, highly intensive, and error prone cooperation between retail and wholesale companies in this sector. All of these factors add complexity to the process of tracking products in the pharmaceutical supply chain. Our research shows that the application of Hyperledger Composer for developing applications running on the Hyperledger Fabric private blockchain network significantly reduces the amount of time needed to develop a blockchain-based software solution.

However, for further improvements to the presented solution, the Hyperledger Fabric platform needs to be used directly on the infrastructure level. Therefore, our future work will include implementing the complete solution directly on the Fabric, using the Go language for writing chaincode. This will allow greater control over consensus rules, endorsement policies, security, and business logic of the system. Hospitals will also be added as a special participant in the network. The web application, which now supports only CRUD operations, will also support custom business transactions. In the current version of the solution, only the backend component contains validation. Therefore, validation and improvement of the frontend design will be also part of our work to come.

ACKNOWLEDGMENT

The research reported in this paper is partly supported by the Ministry of Education and Science of the Republic of Serbia, projects III44010 (2011-2018), III44006 (2011-2018), and ON174026 (2011-2018).

REFERENCES

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, October, 2008.

[2] Wikipedia, Proof of work, available at: https://en.bitcoin.it/wiki/Proof_of_work, last accessed 30/8/2017.

[3] Wikipedia, Proof of stake, available on: https://en.wikipedia.org/wiki/Proof-of-stake, last accessed 30/8/2017.

[4] Fabric Composer Documentation, available at: https://hyperledger.github.io/composer/, last accessed 20/9/2017.

[5] The difference between a Private, Public & Consortium Blockchain, available at: http://www.blockchaindailynews.com/The-difference-between-a-Private-Public-Consortium-Blockchain_a24681.html, last accessed 21/9/2017.

[6] R. C. Merkle, Protocols for public key cryptosystems, in *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pp. 122-133, April, 1980.

[7] Hyperledger Fabric Documentation, available at: https://hyperledger-fabric.readthedocs.io/en/latest/, last accessed 24/9/2017.

[8] M. Castro and B. Liskov, Practical Byzantine fault tolerance, in *Proc. 3rd OSDI*, pp. 173–186, February, 1999.

[9] L. Lamport, R. Shostak, M. Pease, The Byzantine Generals Problem, *ACM Transactions on Programming Languages and Systems*. 4(3): pp. 382-401, July, 1982.

[10] Nick Szabo, *Smart Contracts: Building Blocks for Digital Markets*, available at: www.fon.hum.uva.nl, last accessed 5/8/2017

[11] Hyperledger Project, dostupno na: https://www.hyperledger.org/, last accessed 24/9/2017.

[12] Ethereum, available at https://www.ethereum.org/ , last accessed 20/1/2018.