International Journal for Quality Research 17(3) 651–664
ISSN 1800-6450

Vladimir Milićević [1]
Nemanja Zdravković
Jovana Jović

Article info:
Received 09.04.2022.
Accepted 17.04.2023.

UDC – 005.6
DOI – 10.24874/IJQR17.03-02

ON THE SELECTION OF SUITABLE BLOCKCHAIN TECHNOLOGIES FOR SUPPLY CHAIN MANAGEMENT

Abstract: *In the ever-expanding field of software engineering, the challenge of trust between different users and stakeholders of a software solution is becoming more prevalent. Software end users aim to be fully certain that the solution and/or service is in accordance with their requirements, with no deviations in quality of the selected or ordered solution and/or service. All actors in such a system are incentivized to secure and protect their own data, which is ultimately stored, processed and transferred within such a software system. In this paper, we demonstrate the applicability of an innovative approach to data storing and processing, and information exchange by implementing blockchain technologies, presented in a supply chain management case study. After selecting the appropriate technology, we focus on the security aspects which blockchain technology provides for our use-case. Finally, we test the model with transparency and information integrity hypotheses, network security challenges, and inappropriate actor behavior.*

Keywords: *Blockchain, Ethereum, Hyperledger, Supply Chain Management, Transparency, Trust.*

1. Introduction

Blockchain technology (BCT), while still being a novel technology, is gaining more attraction in different fields products and services based on modern software solutions with intelligent mechanisms. Blockchain is already being applied for numerous applications in different domains as a decentralized approach to the development of software which is resistant to modification and malicious behavior, without the need for a trusted party. We can therefore state that BCT is a distributed ledger which is append-only, cryptographically secure from unauthorized access and revisions (Nakamoto, 2009).

This novel, yet disruptive technology has found its application in fields such as, but not limited to (Chen et al., 2018):

- Cryptocurrencies,
- Healthcare,
- Insurance systems,
- Advertising systems,
- Copyright protection,
- Energy systems,
- Internet-of-Things (IoT),
- Banking,
- Social networks.

The application of this type of technology has the goal to increase the level of user satisfaction through a higher order of trust in the software solution which respects all users' privacy with regards to legal

[1] Corresponding author: Vladimir Milićević
Email: vladimir.milicevic@metropolitan.ac.rs

regulations in the application domain (Bayón, 2019).

The aim of this paper is to analyze and discuss the selection of the appropriate BCT, as well as its applicability to ensure information in supply chain management (SCM) (Tribis et al., 2018). Information systems for SCM do exist, which can solve some of the logistic problems using traditional methods, e.g. applying a centralized or partially distributed software architecture. Using this approach, SCM systems (in a digital form, using commercially bought software or developing a custom software solution) offer business advantages when compared to a management system with no computer-aided support.

These advantages obtained by applying modern software solutions include:

- Easier information transport using applications and server calls within the SCM system as a whole,
- Centralized monitoring the state of the supply chain and its history,
- Ensured data integrity and confidentiality (with the use of encryption and hashing), since a digital signal is more secure than a paper trail,
- Easier and potentially less expensive networking of small and medium-sized enterprises (SMEs) into a supply chain system.

An example of traditional information technology-based (IT-based) SCM system is shown in Figure 1.

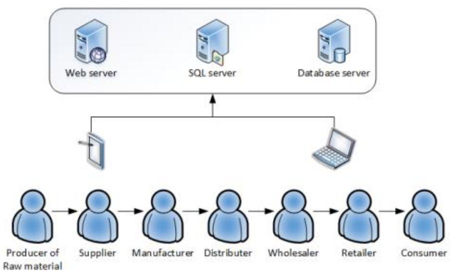

Figure 1. Traditional IT system for SCM.

In such a centralized SCM, all information is sent to the centralized component which stores and forwards that information within the supply chain. However, several issues exist when taking this traditional approach:

- A lack of transparency exists for all actors and/or end users of the supply chain,
- Several potentially useful supply chain information can be ignored,
- Harder tracking mechanism of human errors in the supply chain or in business partner transactions,
- Information flow is limited and overall slower due to a lack of trust between business partners, and instead of being potentially interconnected, partners create of co-called information islands.
- Automated liability tracking.

A question arises: is it possible to securely upgrade existing SCM systems and their method of functioning, and to create such a system that is sustainable and feasible in a business sense? We turn to BCT for a possible solution.

2. The choice of blockchain technology

Prior to developing a software solution which would incorporate BCT for storing and conduction business transactions, a development team has to research in detail various blockchain concepts, technologies and tools, which would enable tom to connect business models with modern technology. It should be noted that the choice of the most appropriate BCT is crucial in order to maximize business interactions after implementing the software solutions. A plethora of BCTs exist; however, for SCM, Ethereum and Hyperledger are the two solutions are most suited, and it is important to choose the solution which fits most, if not all, of the software requirements.

For the choice of the most appropriate BCT it is necessary to identify several criteria. The following measures act as criteria for choosing BCT for a SCM system:

- The purpose of the BCT itself,
- Confidentiality,
- Network privacy type,
- Consensus mechanism applied,
- Programming language support,
- Cryptocurrency support.

Table 1 highlights the main differences between Hyperledger and Ethereum BCTs compared over the selected criteria.

Table 1. Main differences between Hyperledger and Ethereum blockchains.

Characteristic	Hyperledger	Ethereum
Purpose of the BCT	Preferred for B2B commerce	Preferred for B2C commerce and general purpose as well
Confidentiality	Confidential transactions	Transparent
Type of network	Private	Private or Public
Consensus mechanism	Several, no mining needed for some mechanisms	Proof-of-Work and Proof-of-Stake, mining needed
Programming languages	GO, Java	Solidity, smart contracts
Cryptocurrencies	No support	Ether

Apart from the chosen criteria, it is necessary to conduct additional analysis and discussion in order to clearly identify Ethereum and Hyperledger technologies themselves.

Ethereum presents a public distributed blockchain network based on open-source principles. It allows to develop distributed applications on top of the Ethereum layer, called daps, with the use of functionalities which are called smart contracts. As of writing this paper, a large number of smart contract applications exist (Kosba et al., 2016, Idelberger et al., 2016, Alharby et al., 2017). Ethereum was developed by Vitalik Buterin as an upgrade to the existing core blockchain concept (Metcalfe, 2020), modifying the Bitcoin protocol in order to enable support for applications that can be developed without the need for cryptocurrencies. The main contribution of this platform is the development of smart contract concepts, which can be viewed as code that is executed in the network itself. Therefore, this platform enables developers to write applications with decentralized support. Due to its public nature, anyone can access the Ethereum blockchain and become a node on the network. Whereas Ethereum is still viewed as a cryptocurrency blockchain, only second to Bitcoin, Hyperledger offers more freedom to its developers to use the Hyperledger platform to build personalized blockchain networks which meet specific business models. Hyperledger is maintained by the Linux Foundation, which also focuses on open source BCTs. Hyperledger today refers to an umbrella term that encompasses multiple distributed ledger technologies, as well as libraries and tools (Milićević et al., 2021). A simple visual description of the two BCTs is given in Figure 2.

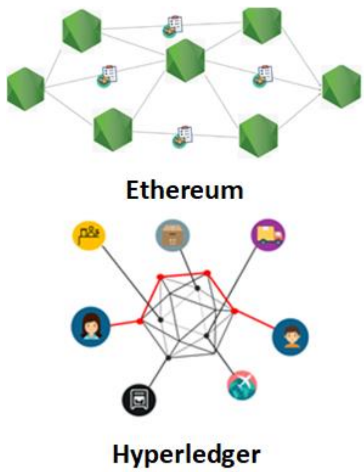

Figure 2. A simple visual description of Ethereum and Hyperledger

Continuing the discussion regarding the similarities and differences between Ethereum and Hyperledger, we highlight specific factors in order to choose the appropriate technology for SCM. Ethereum runs smart contracts on the so-called Ethereum Virtual Machines (EVMs) for decentralized applications meant for a larger public. Hyperledger, on the other hand, covers BCTs for direct business model support. It is designed to support adjustable component implementation which allow for a high level of confidentiality, protection and scalability. In addition, Hyperledger has a modular architecture and allows high flexibility in its use, and therefore, by this criterion seems more adequate for the use in SCM. Concerning transparency, in Ethereum, all transactions are available to all participants in the network, while in Hyperledger, transactions are confidential and accessible only to those nodes that are authorized with a specific encryption key to access the transaction. Furthermore, Hyperledger's privacy allows this type of network to be accessed by only those actors with predefined authorization.

Decisions in most blockchain networks are made using some type of consensus mechanism. Ethereum uses mining-based mechanisms such as Proof-of-Work (PoW) and Proof-of-Stake (PoS) (Sarkar, 2020). This, in turn, means that all nodes in the network have to achieve a consensus over all transactions to make them valid. Hyperledger has precise control over consensus and a limited access to the transactions, resulting in better overall privacy, as well as better scalability. Hyperledger supports more programming languages, such as GO, Java, JavaScript, when compared to Ethereum's support only for Solidity. When developing complex software solutions for SCM, this language flexibility allows the developers to easier adopt Hyperledger for their go-to blockchain-based software solutions.

Finally, Hyperledger does not need support for cryptocurrencies as a method for executing transactions. Ethereum, on the other hand, uses the cryptocurrency Ether, to "pay" for every executed transaction. A Hyperledger-based blockchain solution can be developed to, if needed, incorporate tokens as a means to "pay" for certain transactions.

Based on the discussion presented above, we have concluded that Hyperledger offers more overall flexibility to develop blockchain-based solutions for a SCM system. In the next Section, we provide a more detailed overview on one of Hyperledger's main projects, Hyperledger Fabric, and we model use-cases for SCM.

3. Overview of the selected technology with given use-cases

Blockchain technologies such as Hyperledger, are often referred as private or permissioned blockchains. Furthermore, these types of technologies are also called distributed ledger technologies (DLTs) to distance them from public, cryptocurrency-based blockchains. With Hyperledger Fabric, it is possible to construct a private network with fast transactions and fast smart contract execution, paired with specific approval rules, for a group of organizations which needs to handle transactions on a distributed, transparent and automated manner. In this paper, we test Hyperledger Fabric on the SCM use-case.

Hyperledger Fabric consists of actors and nodes, which are (preferably) distributed across different actors' i.e. organization's servers. Apart from actors and nodes, the network consists of distributed smart contracts which allow and provide, with regards to data protection, created functionalities and activities to be handled over specific rules and use cases. Actors are certified, known, and allowed to make transaction within the created and deployed Hyperledger Fabric network. A diagram which shows an example of user, network components and functionalities is given in Figure 3.

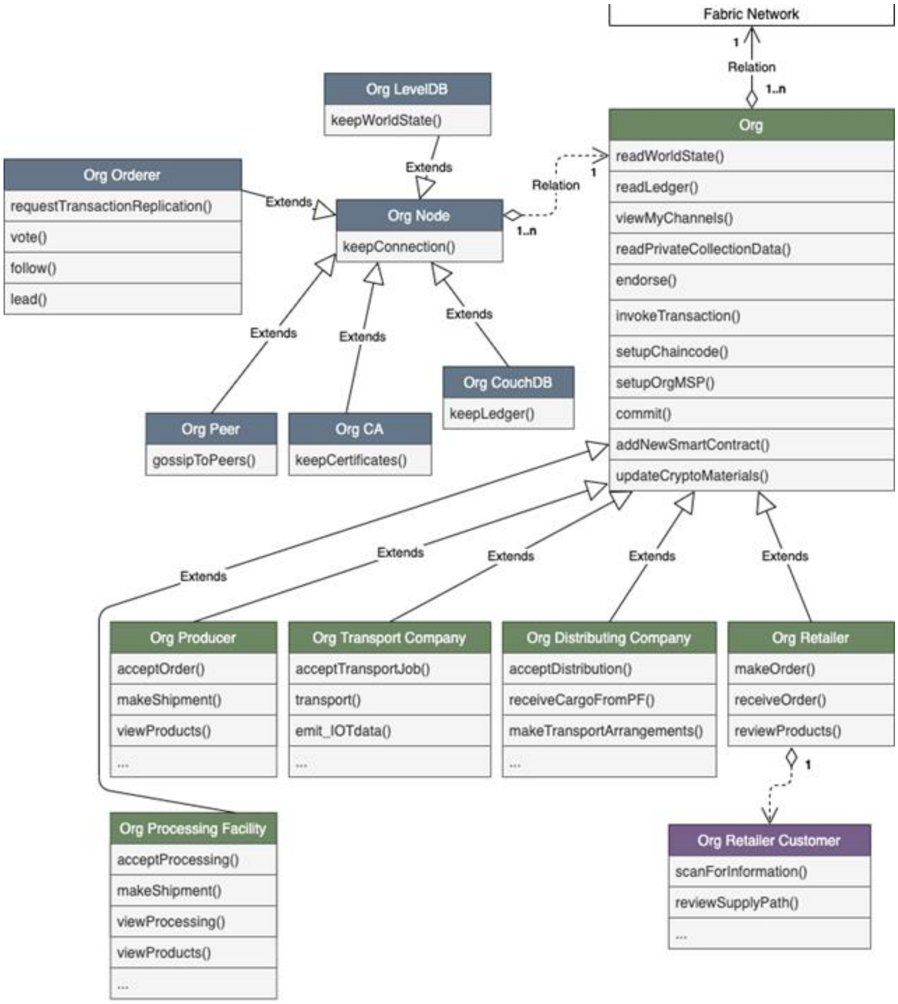

Figure 3. An example of components, actors and their functions within a Hyperledger network

From the diagram in Figure 3 we can observe that Hyperledger Fabric can be implemented with an object-oriented programming language such as Java. Hyperledger Fabric provides a Membership Service Provider (MSP), which is used to identify a network user.

The MSP can also limit certain functionalities to a node or organization within the network, using access control lists (ACLs) or access control modules (ACs) in operating systems. We can distinguish two types of MSP – the former is the local MSP component and the latter is the channel MSP component.

The local MSP is a security module which can be used for defining a node's or applications' single administrator within the Hyperledger Fabric network, or to determine all administrators of a node or application as well. The channel MSP component is used to identify users of a specific channel and to link those identities with user's permissions. In such a manner, the channel MSP can represent the organization's public keys and/or certificates. Every organization within the network has an MSP component in their channel configuration in which transaction can be executed.

Hyperledger Fabric also uses chaincode. Namely, when deploying the network, a co-called system chain is created which defines basic functionalities for a specific network use-case. Besides a basic smart contract, it is possible to make additional smart contracts to expand the functionalities of the network. An example of smart contracts functions is given in Figure 4.

```
*/
async transportShipment(ctx, orderId) {⋯
}

/**⋯
async receiveShipment(ctx, orderId) {⋯
}

/**⋯
async queryOrder(ctx, orderId) {⋯
}
```

Figure 4. An example of functions in a smart contract

The Hyperledger Fabric DLT is comprised of actors which, based on the consensus mechanism, are able to execute transaction on the network. All actors represent business entities, such as organizations or companies. As an example, for an organic food products SCM system use-case, the actors can be the following:

- Producer – a company that manages the SCM of fresh products,
- Retailer – a market that sells organic products,
- Organic restaurant – a restaurant with offers for producers,
- Transport – a company which transports organic products from the producer to the restaurant, to the organic food processing facility, or to the distribution center,
- Distribution center – a company which, at some point in the supply chain, buys the products and distributes them to the supply chain end users – retailers or restaurants,
- Organic food processing facility – an actor tasked with some sort of processing organic food into a new product,
- Regulator – a regulatory body or administrative organization tasked with monitoring all transactions regarding food production, restaurant, retailer or facility delivery,
- Customer – a restaurant guest which rates the quality of served organic food, as well as overall dining experience and information regarding product history within the supply chain.

The nodes in the network are network components of the Hyperledger Fabric needed to efficiently and securely send all transaction, as well as receive and process all transactions within the network. The nodes types are as follows:

- Peer Node – a node which is interconnected with other nodes of the same type,
- LevelDB Node (World State Ledger Node) – a node which stores the organization base i.e. current state of all network information,
- CouchDB Node (Blockchain Ledger Node) – a node which stores the organization base with all network transactions (both successful and unsuccessful), with all parameters,
- Certificate Authority (CA) Node – a node which stores the certificates (public and private keys) of all organizations and its employees which interact with the Hyperledger Fabric network SCM,
- Ordering Peer Node – nodes which are involves with transactions and blocks and which validate adding a block to the ledger.

A single node can have multiple functions, e.g. a node can be both an endorsing node and a committing peer node.

Hyperledger Fabric has a specific manner for storing information. Namely, it has two databases, a base with transaction (on the blockchain itself) and a base with metadata (world state database) which holds the current value of the ledger, as shown in Figure 5 (Hyperledger Fabric, 2020).

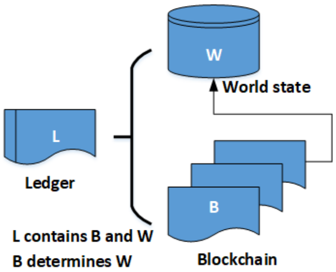

Figure 5. Databases W and B, LevelDB and CouchDB

Prior to presenting the proposed network structure, it is necessary to discuss the consensus mechanism and confidentiality policies.

We start with the following assumptions:

- The proposed application is a portal to a Hyperledger Fabric network with SCM support,
- The network is deployed with all configurations, crypto-materials (certificates) and all organizations have a smart contract installed for different network roles,
- All components are interconnected.

The approval policies are therefore as follows:

- Starting at the portal application, actors can only send a transaction proposal. A transaction is a function call to the installed smart contract,
- Validating nodes, i.e. endorsing peers, check if the call is properly formed (by checking with standardized input data), if the transaction is already sent (to check for replay attacks) and to check if the digital signature is valid (MSP check),
- With the help of the input parameters of the called function, endorsing peer nodes simulate the existence of the function over the current state database. Afterwards, a proposal response is sent for that transaction containing the added information in the database for that same transaction coupled with the digital signatures of the validating nodes. In this moment, the transaction is only simulated an no values are modified or added to the node databases, as well as on the world state database.
- The application verifies the validator nodes' signatures and checks if the proposal response from them are the same. It also checks if the allowance policies were met.
- This type of transaction with the endorsement responses, is hence broadcasted to the nodes which collect the approved transactions, sort them by timestamps, and create a new block.
- These blocks are therefore sent back to all the nodes and the transactions within are checked again. The allowance policies and current states are being checked in this step. The transactions in the block are now marked for valid or invalid.
- Finally, the ordering service is been executed. The ordering service is a group of nodes from all parties that are included in the organization of all broadcasted transactions within a time period. Transactions are needed to be organized in blocks in chronological order, i.e. sorted by the time of being in the network.

Starting from version 2.0, Hyperledger Fabric uses the RAFT consensus mechanism. The details of this mechanism is beyond the scope of this paper; however, it is a Crash Fault Tolerant (CFT) mechanism based on the random selection of a leader node and multiple follower nodes. The main task of this algorithm is transaction replication. An example of interconnected nodes, with the RAFT mechanism in the center, for our SCM use-case is shown in Figure 6. Anchor peers (APs) are nodes which connect to the nodes of other parties. Connections of a single organization are shaded.

Data confidentiality within the network itself is achieved Private Data Collections (PDCs). This method is achieved in the following steps:

- Firstly, private information is sent only to those organization which need that information,
- Secondly, this information is passed through the allowance policies as a hash value of the private information,
- Finally, organizations have private information shared among themselves, and in this moment a public transaction can be made across the whole network, coupled with the hashes of private information. The hashed are therefore distributed among all nodes, while private information (in plain text form) is distributed only to certain organizations and nodes.

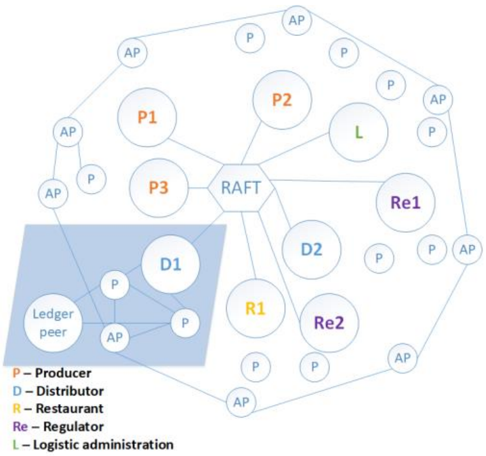

Figure 6. Ordering peer nodes and peer connection for the SCM use-case

4. Use-case

The application itself is very complex and it is not possible to demonstrate all the elements of implementation in one paper. All the most important aspects of the analysis and design of blockchain technology are described as an use case of a the supply chain of organic products. In the following, it is important to adequately demonstrate the most important elements of implementation based on the described elements of analysis and design.

In the beginning, it is necessary to make appropriate integration of the recommendations for identifying the supply chain components of the GS1 Standard (www.gs1yu.org) for fresh food in supply chain management which states that all commercial entities should have a Global Location Number (GLN).

Table 1. Title (Times New Roman, 10pt, align Left, single spacing)

Components	GS1 ID	Description
Geographical location of organic food cultivation	GLN	Exact location of the plot
Products and containers during transportation	GTINs (Global Trade Item Numbers), SSCCS (Serial Shiping Container Codes)	Identification of organic products and transporters
Information broadcasting	GTIN i GLN (Global Location Number)	Blockchain network as a replacement for EDI standard

In order for a product to begin its journey through the supply chain, actors first forward bids to each other to begin commodity transactions. One cycle with all the steps of the product journey through the supply chain follows:

1. A retail store measures its product list and requires the purchase of a new quantity.

Makes and sends an inquiry (for business cooperation) to the distribution center.

2. The distribution center reviews the request and accepts it. Procurement of organic food is required.

3. The distribution center requires the supply of products. Makes and sends an inquiry to the organic food production center.

4. The organic production center reviews the request and accepts it. It is necessary to procure fresh organic food for the production of products.

5. The Center for the Production of Organic Food requires the supply of organic food. Makes and sends an inquiry to the manufacturer (company).

6. The company reviews the request and accepts it.

7. At the same time, the restaurant requires a new order of fresh organic food from the producer.

8. The restaurant sends an inquiry to the company.

9. The company reviews the request and accepts it.

10. Prior to accepting the offer, the company has set its products as purchase offers, to which the offers of the restaurant or center for the production of organic food are tied. For each product or series of products, offer and quantity of products - the regulatory body checks the product and gives permission to sell a particular product or series of products. Documents confirming the validity of the product and the offer are forwarded from the company and the center for the production of organic food further in the supply chain.

11. All bids have a time limit for delivery as well as additional IOT sensor limits related to the transport of goods from actor A to actor B. These IOT sensor limits may be restrictions on the temperature and frequency of GPS broadcasts of containers (that stores and contains products) transit.

12. The company creates the source resource and from that product actually starts the journey through the supply chain in the other direction, towards the retail store or towards the restaurant.

13. The company sends an offer to the transport company for the job of transporting the container with organic food to the restaurant.

14. The transport company reviews the offer and accepts it. The restaurant received information about the transport by broadcasting through the network and expects the transport to arrive by the transport company.

15. The transport company broadcasts information of all IOT sensors on the containers during transport and completes the transfer of goods when it reaches the restaurant.

16. The restaurant confirms that the goods have arrived and checks the condition of the container and the quantity of goods. The transaction between the restaurant and the company was successfully completed.

17. Restaurant guests can have the option to scan the QR code on the account of the ordered food to review the path of the food from the manufacturer to the restaurant.

18. In addition, the company sends an offer to the transport company (same or different) for the job of transporting the container to the center for the production of organic food.

19. The transport company reviews the offer and accepts it. The Center for the Production of Organic Food received information on transport by broadcasting through the network and expects the transport to arrive from the transport company.

20. The transport company broadcasts information of all IOT sensors on the containers during transport and completes the transfer of goods when it reaches the center for the production of organic food.

21. The Center for the Production of Organic Food confirms that the goods have arrived and checks the condition of the

container and the quantity of the goods. The transaction between the organic food production center and the company was successfully completed.

22. The Center for the Production of Organic Food sends an offer to the transport company (the same or another) for the job of transporting the organic food container to the distribution center.

23. The transport company reviews the offer and accepts it. The restaurant received information about the transport by broadcasting through the network and expects the transport to arrive by the transport company.

24. The transport company broadcasts information of all IOT sensors on the containers during transport and completes the transfer of goods when it reaches the distribution center.

25. The distribution center confirms that the goods have arrived and checks the condition of the container and the quantity of goods. The transaction between the organic food production center and the distribution center was successfully completed.

26. The distribution center sends an offer to the transport company (same or different) for the job of transporting the organic food container to the retail store.

27. The transport company reviews the offer and accepts it. The retail store received information about the transport by broadcasting through the network and expects the transport to arrive by the transport company.

28. The transport company broadcasts information of all IOT sensors on the containers during transport and completes the transfer of goods when it reaches the retail store.

29. The retail store confirms that the goods have arrived and checks the condition of the container and the quantity of the goods. The transaction between the distribution center and the retail store was successfully completed.

30. One cycle of the supply chain to the restaurant and to the retail store has been completed.

The following assumptions and limitations of the scenario are discovered:

- The entire supply chain is located in one administrative region or country.
- IOT sensors for transmitting information during the transport of goods are considered to have been tested by a third party or a company in charge of this purpose, which is not in the model network.
- The regulatory body is an external actor that must verify and issue a certificate of verification of the products made to the actors - the company and the center for the production of organic food - because they grow or make new products in the system.
- In case of return of goods, in this case food or products, actors receiving goods from a previous actor in the system, it is assumed that the inspection by an independent expert has already been performed, after reporting that the goods or container packed or contaminated . Therefore, the scenario will not show the process of returning the goods.
- There are types of nodes for connecting with other organizations, executing and storing blocks, approving transaction proposals and for editing services (anchor, commiting and endorsing, ordering nodes). Species are not marked individually, but each organization has at least one node for each node type.
- All nodes, channels, smart contracts are raised and installed online. Smart contracts are written in the Java programming language on Hyperledger Fabric. Actor functions are functions implemented in smart

contracts.

- Organizational databases are those recommended by the Hyperledger Fabric project, LevelDB and CouchDB.

The actors in the network, arranged by channels, are shown in Figure 7, which is an excellent basis for completing the blockchain network implementation model to support the supply chain process.

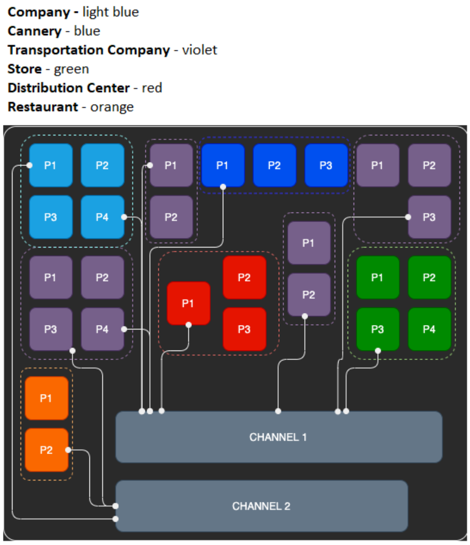

Figure 7. Nodes and channels of actors

Finally, the presentation ends with a demonstration of the flows of transactions that take place within the described blockchain network. Transaction flows are described in the right graph in Figure 8. By adding components: SME actor components, CA component actors, smart contract layout, channel database layout, and appropriate labels for each actor covered by the scenario, the final version of the blockchain network implementation model is obtained and shown in Figure 9.

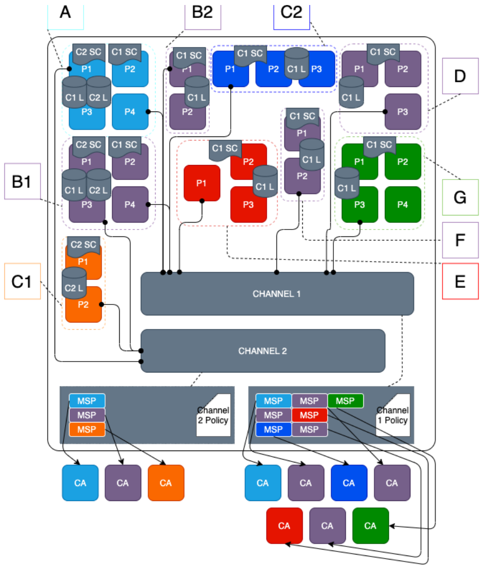

Figure 8. Final model

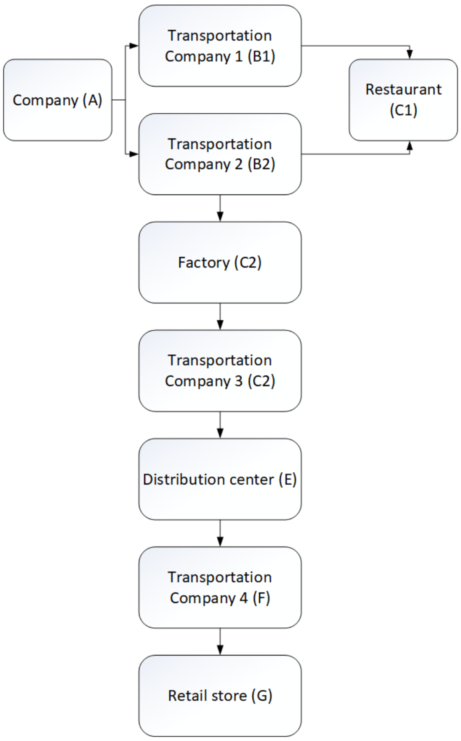

Figure 9. Transaction flow.

5. Conclusion

The aim of this paper is analysis and discussion of the choice of optimal blockchain technology for modern storage, processing and distribution of data in information systems to support supply chains.

Two approaches were compared:

- Ethereum,
- Hyperledger, i.e. its distribution Hyperledger Fabric.

The selection is based on specific criteria of blockchain technology:

- The purpose,
- Confidentiality,
- Type of network,
- Support for consensus mechanisms,
- Support in the form of programming languages,
- Support for cryptocurrencies.

According to all criteria, except support for cryptocurrencies as the optimal sollution to support the supply chain, Hyperledger Fabric was identified and further analysis and discussion was conducted around it.

A special part of the paper is the case study on the introduction of blockchain technology, based on the Hyperledger Fabric platform, to support the supply chain of organic products. The following are described in detail:

- Use case scenario,
- Actors and channels in the network,
- Method of selection and application of the consensus mechanism algorithm,
- Network topology,
- Intra-network transaction flows.

This paper omitted the final definition of the model, which are standard diagrams, well known and present in a huge number of papers, as well as project documentation, such as sequential diagrams for different application scenarios within such information system.

It is certain that this approach has a lot of room for improvement in terms of a more efficient and effective final software solution. It is a very young technology and due to the lack of adequate training, as well as literature, researchers and developers are largely left to experiment and test different scenarios. However, this is an extremely growing technology that will certainly bring numerous benefits to all users and participants of the blockchain network, and the key is certainly to build trust between all participants in the network.

Acknowledgment: This paper was supported in part by the Blockchain Technology Laboratory at Belgrade Metropolitan University, Belgrade, Serbia, and in part by the Ministry of Education, Science and Technological Development, Republic of Serbia (Project III44006).

References:

Alharby, M., Aldweesh, A., & Van Moorsel, A. (2018, November). Blockchain-based smart contracts: A systematic mapping study of academic research (2018). In 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB) (pp. 1-6). IEEE. doi: 10.1109/ICCBB.2018.8756390

Bayón, P. S. (2019). Key legal issues surrounding smart contract applications. KLRI Journal of Law and Legislation, 9(1), 63-91. doi: 10.2139/ssrn.3525778

Chen, W., Xu, Z., Shi, S., Zhao, Y., & Zhao, J. (2018, December). A survey of blockchain applications in different domains. In Proceedings of the 2018 International Conference on Blockchain Technology and Application (pp. 17-21). doi: 10.1145/3301403.3301407

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016, July). Evaluation of logic-based smart contracts for blockchain systems. In International symposium on rules and rule markup languages for the semantic web (pp. 167-183). Springer, Cham. doi: 10.1007/978-3-319-42019-6_11

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In 2016 IEEE symposium on security and privacy (SP) (pp. 839-858). IEEE. doi: 10.1109/SP.2016.55

Metcalfe, W. (2020). Ethereum, smart contracts, DApps. Blockchain and Crypt Currency, 77. doi: 10.1007/978-981-15-3376-1_5

Milićević, V., Jović, J., & Zdravković, N. (2021). An overview of Hyperledger blockchain technologies and their uses. In *Proc. Of the 11th International Conference on Information Society and Technology (ICIST 2021)* (pp. 62-65).

Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System-Satoshi Nakamoto-The Bitcoin Whitepaper.

Sarkar, P. (2019). A New Blockchain Proposal Supporting Multi-Stage Proof-of-Work. Cryptology ePrint Archive.

Tribis, Y., El Bouchti, A., & Bouayad, H. (2018). Supply chain management based on blockchain: A systematic mapping study. In MATEC Web of Conferences (Vol. 200, p. 00020). EDP Sciences. doi: 10.1051/matecconf/201820000020

Vladimir Milićević
Faculty of Information Technologies, Belgrade Metropolitan University
Belgrade, Serbia
vladimir.milicevic@metropolitan.ac.rs
ORCID 0000-0002-5587-2717

Jovana Jović
Faculty of Information Technologies, Belgrade Metropolitan University
Belgrade, Serbia
jovana.jovic@metropolitan.ac.rs
ORCID 0000-0002-4204-0233

Nemanja Zdravković
Faculty of Information Technologies, Belgrade Metropolitan University
Belgrade, Serbia
nemanja.zdravkovic@metropolitan.ac.rs
ORCID 000-0002-2631-6308