Proceedings Paper:
Butean, A, Pournaras, E, Tara, A et al. (2 more authors) (2020) Dynamic Consensus: Increasing Blockchain Adaptability to Enterprise Applications. In: Proceedings of the 9th Computer Science On-line Conference 2020. CSOC 2020: Applied Informatics and Cybernetics in Intelligent Systems, 15 Jul 2020, Zlin, Czech Republic. Springer International Publishing , Cham, Switzerland , pp. 433-442. ISBN 9783030519735

https://doi.org/10.1007/978-3-030-51974-2_41

Dynamic Consensus: Increasing Blockchain Adaptability to Enterprise Applications

Alex Butean[1], Evangelos Pournaras[2], Andrei Tara[1], Hjalmar Turesson[3], Kirill Ivkushkin[4]

[1] Lucian Blaga University of Sibiu, 10 Victoriei, Sibiu, Romania
alex@butean.com, research@andreitara.com

[2] University of Leeds, Leeds LS2 9JT, United Kingdom
e.pournaras@leeds.ac.uk

[3] York University, Toronto, 4700 Keele Street, Canada
hturesson@schulich.yorku.ca

[4] Insolar Technologies GmbH, Hinterbergstrasse 49, Steinhausen, Canton of Zug, Switzerland
kirill.ivkushkin@insolar.io

Abstract. Decentralization powered by blockchain is validated for its capability to build trust like no other computational system before. The evolution of blockchain models has opened new use-cases that are becoming operational in many industry fields such as: energy, healthcare, banking, cross-border trade, aerospace, supply chain, and others. The core component of a decentralized architecture is the consensus algorithm - the set of rules that ensures an automated and fair agreement between the actors in the same network. Classic consensus algorithms are tailored to solve specific problems, but in an open ecosystem, each business case is unique and needs a certain level of customization. This paper introduces a new meta-consensus model called Dynamic Consensus, an architecture extension that allows multiple, complementary, consensus algorithms to run on the same platform. While classic consensus mechanisms are more appropriate for public or private systems (narrow set of rules), a dynamic approach would fit better for federated business consortiums (more rules and higher need for adaptability). The model is illustrated and analyzed as an ongoing experimental feature that can be added to enterprise blockchains designed to operate in cross-domain environments.

Keywords: Decentralized System, Blockchain, Consensus, Enterprise

1. Introduction

1.1 Classic consensus models

Proof of Work (PoW) was the first consensus model that applied the Nakamoto consensus [31] for blockchains. The protocol requires each block to contain a solution to a proof of work puzzle (i.e. the PoW) and to point to the previous valid block with the best PoW. Despite its energy inefficiency, variations of this algorithm are used by

many platforms (Bitcoin, Ethereum, etc,) that are implementing fairly simple transaction systems [14]. Even if the algorithm is computationally intensive and slow, it provides a secure and verifiable proof of the entire history of the chain.

Another classic consensus model is the Practical Byzantine Fault Tolerance (pBFT) algorithm, proposed as a solution to the Byzantine Generals' problem [8]. The algorithm is known for transaction finality and attack resistance but works only with a limited number of consensus members since it was designed for leader-based systems that can reduce the quadratic communication complexity of pBFT-like protocols.

The Proof of Stake (PoS) family of algorithms [12] was inspired by the social ecosystem of humans, where the trustworthy ones (risk takers) are entitled to decide over the next state of the network. Ouroboros [19] and Algorand [15] opened the road for Casper [7] and other variations that are following a similar path. Considering the acceptable trade-off between energy efficiency and decentralization, PoS has few proven vulnerabilities [13] and it is considered one of the most balanced consensus methods.

Besides the briefly presented consensus models above, there are many other algorithms and variations that are worth mentioning: Proof of Burn, Proof of Elapsed Time, Proof of Capacity, Proof of Identity and others [3].

From an analytical point of view, several studies [28, 4, 27, 5] have reached an important conclusion: each classical consensus mechanism comes with performance trade-offs, each has its own advantages and disadvantages, and thus, it can perform better or worse, depending on the application context, business perspective and hardware infrastructure constraints.

1.2 Enterprise perspective

If we analyze the range of blockchain models from public to enterprise, we can identify the following focus points:

- transparency was traded for on-demand permission-based access and controlled privacy [36]
- the development and assessment of benchmarks and frameworks are focused on analyzing the existing architectures in order to identify the best choice for a specific context [26, 11];
- the optimization of algorithms is targeting only certain parameters: high-performance [21], efficiency [24], decentralized storage [32]. In this case, a multi-objective optimization would be very hard to perform;
- already functional products are using domain-focused solutions, tailored for individual applications and technologies [20, 2]
- industrial international standardization initiatives are already mature enough to be implemented in the next generation of products [16, 18]

Looking over the above points, we can conclude the following: for multi-organization enterprise networks, there is no easy way to establish a sole consensus mechanism because the need for flexibility is higher than the need for cross-organization communication.

1.3 Practical solutions

Pioneers of enterprise-ready blockchain solutions are already using permissioned consensus methods that are flexible enough to match a large variety of use-cases. For example, Corda R3 is not using block-based ordering, instead, it uses notaries for transaction ordering and timestamping services to reach consensus (validity and uniqueness) [6]. The Hyperledger family clearly states that there is no magic key that opens all doors, that is why various products are using different consensus mechanisms, most of them based on permissioned voting systems: Kafka for Fabric, Proof of Elapsed Time for Sawtooth, Sumergi in Iroha and RBFT in Indy [34].

There are many use-cases, pilot applications or even commercial solutions developed using Hyperledger [29] or Corda [25]. Most of these solutions require a custom adaptation effort, particularly because of these two aspects:

- the virtual machine model needs more components in order to be able to process smart contracts written in high-level programming languages required for a large variety of constraints, rules and integration needs [33, 1];
- the consensus model needs on-demand difficulty control [22] to allow fast adaptation. Such an engine would be able to offer more flexibility for cross-domain, cross-shard [1], multi-organization rules or consortiums [10].

2. Dynamic Consensus

2.1 Starting point

Based on the summarized analysis presented above (classic consensus models, enterprise perspective, practical solutions), this section introduces the concept of Dynamic Consensus, a meta-mechanism meant to increase the flexibility and adaptability of blockchain solutions by leveraging the usage of multiple consensus rules at the same time.

An enterprise network is a private permissioned network with thousands of cross-domain or cross-organization processes that are functioning in parallel. As depicted in Fig. 1, the data exchange is intense and for a realistic scenario, it has to be governed by a continuously changing set of rules.

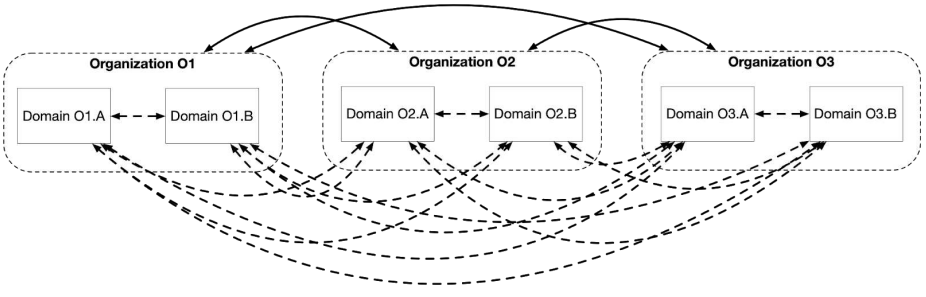

Fig 1. Data exchange in a multi-organization environment

The needs for such a business ecosystem are very hard to reproduce using fixed logic components. Such a network would be cost-ineffective and underperforming with a classic consensus mechanism where all nodes have to be aware in real-time of the entire state space. Also, a single consensus mechanism would probably reduce the parallelism capacity of the system.

2.2 Hierarchical structure

An enterprise network is usually built across a consortium of organizations (companies) aiming for interoperability and data exchange. Each organization has its own domains (departments or sub-organizations) and each domain controls multiple shards (data processing clusters). Each shard is responsible for a specific industrial process and contains several nodes (processing agents). The nodes perform the actual consensus computations. Fig. 2 shows the hierarchical structure of an enterprise network that is designed for a generic consortium.

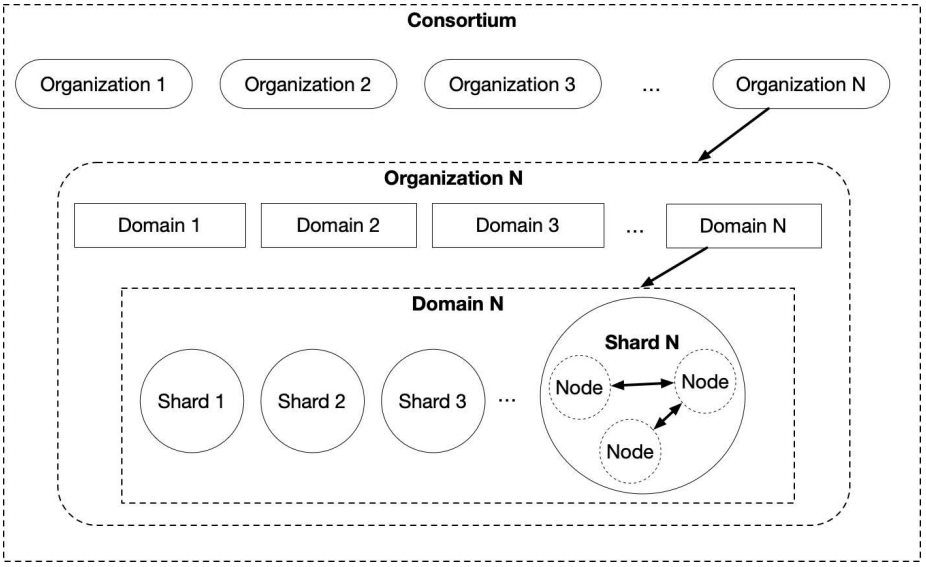

Fig 2. Hierarchical design of an enterprise network

In a blockchain architecture, any data-exchange that affects the decentralized ledger is formally considered a transaction. In the above-presented design, once a transaction is issued, it is fairly easy to identify its origin and target by using the system hierarchy. This is the first step that allows the grouping of transactions based on their source of emittance and their source of impact. In this case, local transactions are processed only at a lower level with a reduced number of nodes, while cross-entity transactions require more decision nodes. This approach saves processing and propagation time and allows the use of an ad-hoc consensus at a lower scale and the use of an overall consensus of all nodes over some global (system-wide) transactions.

2.3 Channels and virtual domains

Side channels, plasma, state channels and other similar mechanisms [35,9], are solutions that keep only relevant information on-chain while the processing is performed off-chain.

In the context of the above-presented consortium design, we propose to use channels as communication rooms where members (nodes) from multiple organizations can subscribe to specific topics. The members of a channel are not dependent on the hierarchical localization and channels are turning into virtual cross-organization domains. Each such domain will be created with the approval of the involved organizations. All operations inside the channel can be processed with a topic-driven consensus algorithm chosen by the participating members (nodes). Fig.3 presents an example of a channel that acts as a virtual domain across organizations.

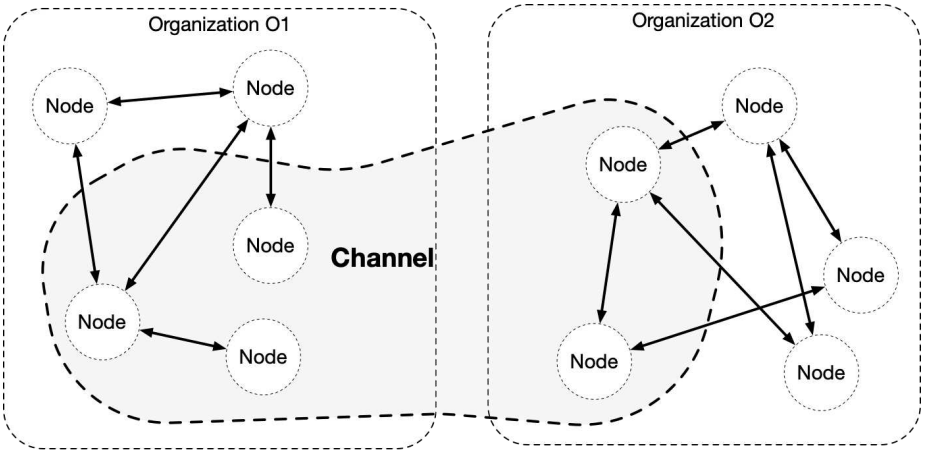

Fig 3. Channel acting as a virtual domain across organizations

2.4 Indexing functions

In the configuration stage, each organization defines the following parameters for its nodes:

- contextual power (C_p). The higher the value, the larger the impact of the node in the decision process. C_p can be considered a low pass filter, where a node with a maximum C_p can be involved in all the decisions across the consortium, while a node with a minimum C_p will be involved only in the least important decisions;
- location interest (L_i). Each location has a unique identifier across the network. Using an array, each node specifies its interest value (boolean) for every specific location in the network (organization, domain, virtual domain)

From a generic perspective, a transaction has the following fields: type(*t*), asset(*as*), source(*so*), target(*ta*), timestamp(*ts*), context(*c*), location effect(*le*). Considering the system-wide constant expiration timestamp(θ) and predicate functions *e(ts)*, *d(so,ta)* defined as:

$$e(ts) = \begin{cases} 0 \; if \; ts \geq \theta \\ 1 \; if \; ts < \theta \end{cases} \quad d(so, ta) = \begin{cases} 0 \; if \; so, ta \notin same \; network \\ 1 \; if \; so, ta \in same \; network \end{cases}$$

The importance function (IMP_{func}) is a classification function that outputs the corresponding consensus algorithm and is defined as follows:

$$IMP_{func}: < T, As, C, L, Ad, Ad, Ts > \rightarrow X$$

$$IMP_{func}(t, as, so, ta, ts, c, le) = classify\ (< t, as, le, d(so, ta), e(ts) >) \text{ where}$$

$X = \{ PoW, PoS, pBFT, \ldots \}$ - contains all classes of alg. supported by the system
$T = \{ t \mid t \in N \;\; and \;\; 0 - register \; organization, 1 - move \; asset, \ldots \}$
$As = \{ as \mid as \in N \;\; and \;\; 0 - no \; asset, 1 - numeric \; asset, \ldots \}$
$C = \{ c \mid c \in N \;\; and \;\; 0 - critical, 1 - business, 2 - \;\; operational, \ldots \}$
$L = \{ le \mid le \in N \; and \;\; 0 - consortium, 1 - organization, 2 - local, \ldots \}$
$Ts = \{ ts \mid ts \in N \;\; and \;\; ts \; is \; the \; timestamp \}$
$Ad = \{ a \mid a \; is \; an \; account \; address \}$

The exact mathematical weights of the classification function are dependent on the configuration parameters of each system. Multiple interactive simulations and relevant training data can reveal optimal values based on hardware specifications, network size and business case. At the system level of each node, there is a shared lookup routing table (Fig. 4) that stores the routing address for all the available consensus algorithms. For each transaction the IMP_{func} determines the associated class of the consensus algorithm. The algorithm address is determined by querying the lookup table for the determined class.

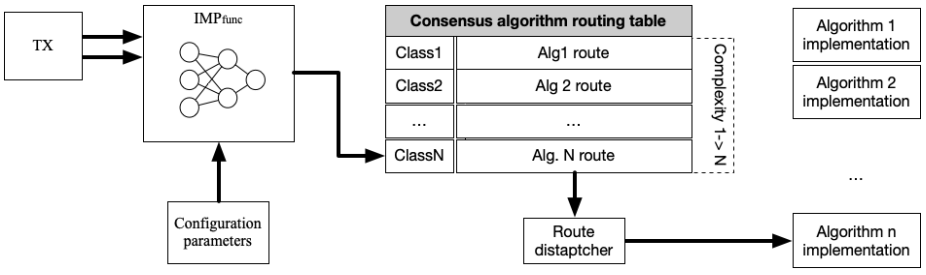

Fig. 4. Indexing table of consensus algorithms

The algorithm dispatch process is dynamically executed at the node level for every visible transaction in the pool. A node decides to participate in a consensus round for a transaction if the following conditions are met at the same time:

- the context(c) of a transaction corresponds to the contextual power of the node (C_p);
- the location effect(le) of a transaction corresponds with the location interest (L_i) of the node

2.5 Overall architecture

Most of the enterprise-oriented blockchain architectures [34, 6, 17] already offer a high level of abstraction. From a software engineering perspective, in order to migrate towards a dynamic consensus capable platform, the static consensus module should be replaced with a meta-consensus component. This component acts as a dispatcher for all transactions. From the perspective of a consortium owner, Fig.5 displays an overall functional architecture.

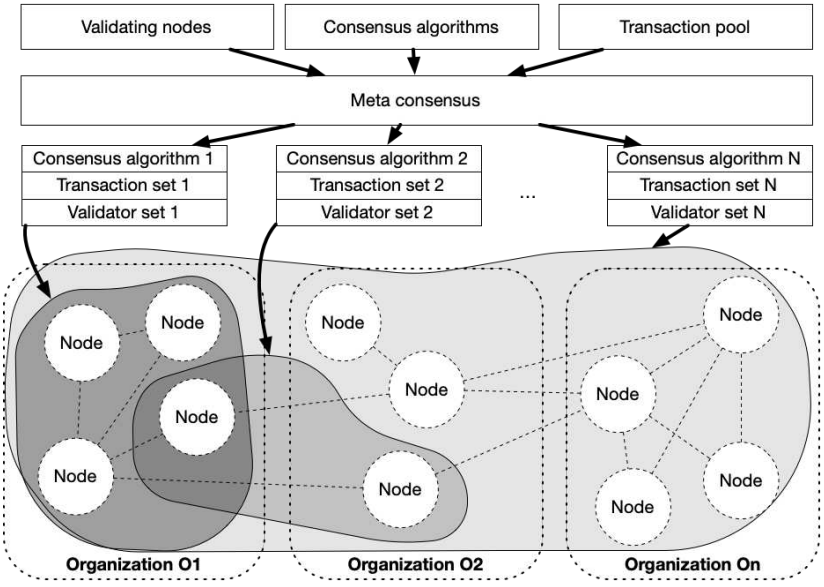

Fig. 5. Dynamic Consensus Architecture

The level of consensus parallelism can be observed in the context of multiple heterogeneous groups that are processing transactions with different consensus algorithms. The most important nodes should run on a hardware configuration with higher security requirements, thus, they can participate in multiple groups at the same time.

2.6 Prerequisite actions for integration

In order to integrate the dynamic consensus model in an existing multi-organization infrastructure, the following guidelines need to be followed :

- the previously described indexing functions are based on a common set of rules that has to be defined at the consortium level. All existing rules will balance the importance, location and context according to application-specific needs. A new rule added to the set has to be validated by the entire network with a top-level voting-based consensus algorithm;
- for reducing the complexity of simulation, the initial state can be: all the nodes receive the same power and the meta consensus is fed with only one entry. Gradually, adding more algorithms and adjusting the power of the nodes should be a context-driven decision;
- the existing static consensus has to be generalized and abstracted in order to be able to swap the modules;
- the host network should activate the support for state decoupling procedures like state sharding [23], layering, etc.

3. Conclusions

The performance of enterprise blockchains is a topic of great interest. Some of the existing platforms are highly adapted to specific needs but are unable to scale and others are highly scalable but not flexible enough for cross-organizational collaboration.

The purpose of this paper is to show that a master consensus algorithm is suboptimal in a multi-organization network. As an substitute, we propose a dynamic consensus model that comes with the following advantages:

- different rules for different transactions based on relevance and impact;
- network consensus (machine-to-machine) can be separated from the business consensus (might require human intervention);
- the bottleneck for parallelism is solved using a hierarchical structure;
- the data exchange in an open collaborative ecosystem can be regulated to match enterprise privacy and security policies;
- from an architectural perspective, the presented model covers the decentralization, consensus and security properties described by the DCS Theorem [30];

Dynamic Consensus is currently being integrated and tested on the Insolar Assured Ledger Platform [17] on a sharded multi-organization infrastructure. Future publications will include comparison sheets between classic static implementations and the proposed dynamic approach.

References

1. Al-Bassam M. , Sonnino A., Bano S., Hrycyszyn D., Danezis G.:Chainspace: A Sharded Smart Contracts Platform", arXiv:1708.03778v (2017)
2. Ali A.A, El-Dessouky I., Abdallah M., Nabih A.: The Quest for Fully Smart Autonomous Business Networks in IoT Platforms. In Proceedings of the 3rd Africa and Middle East Conference on Software Engineering . ACM, NY, USA, pp 13–18 (2017)
3. Anwar H.: 101Blockchain: Consensus Algorithms Blockchain, Available at https://101blockchains.com/consensus-algorithms-blockchain/ (2018)
4. Baliga A.: Understanding Blockchain Consensus Models", Persistent Systems, Available at https://www.persistent.com/ (2017)
5. Ballandies M.C., Dapp M.M., Pournaras E.: Decrypting distributed ledger design-taxonomy, classification and blockchain community evaluation". arXiv preprint arXiv:1811.03419 (2020)
6. Brown R.G: The Corda Platform: An Introduction, Corda R3 Documents (2018)
7. Buterin V., Griffith V.: Casper the Friendly Finality Gadget", ArXiv, 1710.09437v4, (2019)
8. Castro M., Liskov B: Practical Byzantine fault tolerance. In Proceedings of the third symposium on Operating systems design and implementation (OSDI '99). USENIX Association, USA, pp. 173–186. (1999)
9. Coleman J., Horne L., Xuanji L.:Counterfactual: Generalized State Channels, Available at https://l4.ventures/ (2018)
10. Dib O., Brousmiche K-L., Durand A., Thea E., Hamida E..B, Consortium Blockchains: Overview, Applications and Challenges", International Journal on Advances in Telecommunications, vol11no1&2 (2018)
11. Dinh T.T.A.D, Wang J., Chen G., Liu R., Chin Ooi B., Tan K.: BLOCKBENCH: A Framework for Analyzing Private Blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17). ACM, NY, USA, 1085–1100. (2017)
12. Garcia Ribera E. , "Design and Implementation of a Proof-of-Stake Consensus Algorithm for Blockchain", PhD Thesis at Universitat Politècnica de Catalunya (2018)
13. Gazi P., Kiayias A., Russell A.:Stake-Bleeding Attacks on Proof-of-Stake Blockchains, IEEE Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 85-92 (2018)
14. Gervais A., Karame O.K., Wüst K., Glykantzis V., Ritzdorf H., Capkun S.: On the Security and Performance of Proof of Work Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). Association for Computing Machinery, New York, NY, USA, pp. 3–16 (2016)
15. Gilad Y., Hemo R., Micali S, Vlachos G., Zeldovich N.: Algorand: Scaling byzantine agreements for cryptocurrencies, Proceedings of the 26th Symposium on Operating Systems Principles, pp. 51–68 (2017)
16. Gramoli V., Staples M.: Blockchain Standard: Can We Reach Consensus?, IEEE Communications Standards Magazine, Vol. 2, Issue 3, pp. 16-21 (2018)
17. Insolar Team : Insolar Technical Paper (2019) Available at https://insolar.io/uploads/Insolar%20Tech%20Paper.pdf
18. International Organization for Standardization, ISO/TC 307 - Blockchain and distributed ledger technologies, 2016, Available at https://www.iso.org/committee/6266604.html

19. Kiayias A., Russell A., David B., Oliynykov R. (2017) Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In: Katz J., Shacham H. (eds) Advances in Cryptology. Lecture Notes in Computer Science, vol 10401. Springer, Cham (2017)
20. Kim H., Laskowski, M.: Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance, Intelligent Systems in Accounting, Finance, and Management, Vol. 25, Issue 1, pp. 18-27 (2018)
21. Khan N.: FAST: A MapReduce Consensus for High Performance Blockchains. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (BlockSys'18). ACM, NY, USA, 1–6. (2018)
22. Kraft D. Difficulty control for blockchain-based consensus systems. Peer-to-Peer Netw. Appl. 9, 397–413 (2016)
23. Luu L., Narayanan V., Zheng C. ,Baweja K., Gilbert S., Saxena P.: A Secure Sharding Protocol For Open Blockchains", ACM SIGSAC Conference on Computer and Communications Security, pp. 17-30, ACM, NY, USA (2016)
24. Milutinovic M., He W., Wu H., Kanwal M.: Proof of Luck: an Efficient Blockchain Consensus Protocol. In Proceedings of the 1st Workshop on System Software for Trusted Execution (SysTEX '16). ACM, NY, USA, Article 2, pp. 1–6 (2016)
25. Mohanty D., R3 Corda for Architects and Developers:, ISBN-13: 978-1484245316, Apress (2019)
26. Moorsel A.V.: Benchmarks and Models for Blockchain: Consensus Algorithms. SIGMETRICS Performance Evaluation Review (2018)
27. Nguyen G., Kim K.: A Survey about Consensus Algorithms Used in Blockchain, Journal of Information Processing Systems, vol. 14, no. 1, pp. 101-128 (2018)
28. Pîrlea G., Sergey I.: Mechanising blockchain consensus. In Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs . Association for ACM, New York, NY, USA, 78–90 (2018)
29. Shah N.: Blockchain for Business with Hyperledger Fabric, ISBN-13: 978-9388511650, BPB Publications (2019)
30. Slepak G., Petrova A.: The DCS Theorem, arXiv:1801.04335v1, (2017)
31. Stifter, N., Judmayer, A., Schindler, P., Zamyatin, A., & Weippl, E.R. (2018). Agreement with Satoshi - On the Formalization of Nakamoto Consensus. IACR Cryptology ePrint Archive (2018)
32. Tang Y., Zou Q., Chen J., Li K., Kamhoua C.A., Kwiat K., Njilla L., "ChainFS: Blockchain-Secured Cloud Storage", IEEE 11th International Conference on Cloud Computing, pp. 987-990 (2018)
33. Tara A., Ivkushkin K., Butean A., Turesson H. :The Evolution of Blockchain Virtual Machine Architecture Towards an Enterprise Usage Perspective. In: Silhavy R. (eds) Software Engineering Methods in Intelligent Algorithms. CSOC 2019. Advances in Intelligent Systems and Computing, vol 984. Springer (2019)
34. The Linux Foundation, Hyperledger Architecture, Volume 1, Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus, (2017)
35. Vasa V. , Difference Between SideChains and State Channels, Hackernoon (2018) Available at https://hackernoon.com/difference-between-sidechains-and-state-channels-2f5dfbd10707
36. Zhang J.: Kaleido - Permissions & Privacy: Core Elements of an Enterprise Blockchain Available at https://kaleido.io/permissions-privacy-in-enterprise-blockchain/ (2019)