

Endorsement in Hyperledger Fabric Concerning Endorsement Policies

Subjects: [Computer Science](#), [Software Engineering](#)

Contributor: Xuefeng Piao , Hao Ding , Huihui Song

Hyperledger Fabric is one of the most popular, non-profit and enterprise-oriented blockchain platforms of consortium blockchain maintained by the Linux Foundation since 2015. For Hyperledger Fabric, it delivers some key differentiating capabilities over other blockchain frameworks or platforms like modular and configurable architecture, pluggable consensus protocols, etc.

blockchain

latency performance

endorsement policy

Hyperledger Fabric

1. Introduction

Blockchain, initially introduced alongside crypto currencies like Bitcoin ^[1], has risen to fame swiftly in recent years. Its remarkable capacity to establish trust among untrusted parties is a key driver, eliminating the need for a trusted third entity. The widespread application of blockchain technology has reshaped multiple industries both academically and industrially ^[2]. For most blockchain-enabled IoT applications, blockchain serves as a distributed and tamper-resistant ledger for storing IoT data. This effectively bridges the gap in the authenticity and trustworthiness of device data to a certain extent ^{[3][4]}.

At present, an escalating number of blockchain frameworks or platforms like Ethereum, Hyperledger Fabric, Multichain, etc., have become available for development. This is largely attributed to the growing contributions from both corporations and open-source communities ^[5]. Among the numerous blockchain frameworks and platforms, Hyperledger Fabric is one of the most popular, non-profit and enterprise-oriented blockchain platforms of consortium blockchain maintained by the Linux Foundation since 2015 ^[6]. For Hyperledger Fabric, it delivers some key differentiating capabilities over other blockchain frameworks or platforms like modular and configurable architecture, pluggable consensus protocols, etc. In particular, arbitrary smart contracts authored in general-purpose programming languages can be developed and run on Hyperledger Fabric, which realizes the automation of multi-step processes ^[3].

Hyperledger Fabric is now being exploited for multiple IoT scenarios, ranging from healthcare ^[7] to smart grids ^[8]. Furthermore, for latency-sensitive scenarios, applications seeking to implement the corresponding functions on the basis of blockchain also need to consider responsiveness in addition to security and privacy concerns ^{[9][10]}. However, Hyperledger Fabric tends to encounter high bandwidth overhead and latency during use, which prevents it from being applied to latency-sensitive scenarios.

For Hyperledger Fabric, a unique architecture called the Execute-Order-Validate (EOV) model is introduced for transaction processing and execution. The EOV model consists of the following three phases: the endorsement phase, ordering phase and validation phase. The endorsement policy mechanism plays an indispensable role in both phases, which specifies some network peers to confirm, validate and, finally, subsequently append the transaction to the ledger. In short, the endorsement policy is specified for a chaincode when it is approved and committed to the channel. From the perspective of the EOV model, the transaction latency can be seen as arising from those three phases, respectively. Moreover, the major origins of transaction latency can be regarded as the endorsement phase and ordering phase according to the conclusions presented by Thakkar et al. [11] and Kwon et al. [12].

2. Endorsement in Hyperledger Fabric Concerning Endorsement Policies

Despite its rising popularity, scalability remains a key challenge for blockchain technology. High scalability refers to the blockchain system's capacity to efficiently manage and process a substantial volume of transactions and data in order to satisfy the requirements of its users. However, from the point of view of performance metrics, high scalability usually requires a system with high throughput and low latency [13][14][15].

The magnitude of throughput is an important indicator of a blockchain system's ability to process transactions and many researchers have made great contributions to improving the throughput of Hyperledger Fabric. Gorenflo et al. [16] make architectural changes to Hyperledger Fabric, thus reducing computation and I/O overhead during transaction ordering and validation. And the changes are fully plug-and-play. Nakaike et al. [17] analyze the database systems used in Hyperledger Fabric and draw a conclusion that performance bottleneck arises from accesses to the databases that store the latest key-value pairs in the ledger data, indexes to transactions and the update history. Berendea et al. [18] introduce a novel gossip design for Fabric that achieves the simultaneous optimization of the propagation time, tail latency and bandwidth consumption. Such improvement also performs well on high throughput and concurrent applications. Hang et al. [19] propose a transaction traffic control approach based on fuzzy logic and implement it in a smart contract. The blockchain performance is significantly improved by this control approach. Sharma et al. [20] explore Fabric from the perspective of database research and try to migrate certain concepts and technology from the database into blockchain. Their experiments demonstrate that this introduction improves performance by nearly a factor of three.

In IoT applications, high latency stands out as a critical challenge. Presently, the primary objective in blockchain optimization is the minimization of latency. Endorsement policy, identified as one of three major performance bottlenecks of Hyperledger Fabric [11][21], has attracted considerable interest in recent years. Thakkar et al. [11] propose to introduce aggressive caching or parallelization technology for endorsement policy validation, which effectively shortens the time required to verify the endorsement signature of a transaction. In particular, Kwon et al. [12] achieve a substantial improvement in original read transaction processing by distinguishing between read and write transactions in the endorsement phase. Liu et al. [22] then consider how to make the distribution of endorsement tasks more balanced for the endorsement phase of peer nodes in order to improve the performance

of a blockchain system in the case of high concurrency. Shammam ^[23] et al. investigate the role endorsement plays in Hyperledger Fabric's consensus and involve it in their proposed control model, overcoming traditional centralized access control issues to a certain extent. Qin et al.'s study ^[24] is also concerned with the significance of endorsement policy in designing a control scheme and make dynamical modifications to the endorsement protocol according to their proposed user credibility incentive mechanism. The modifications work well for the protocol. Wang et al. ^[25] acquire, through their experimental results, proof that Hyperledger Fabric exhibits good scalability in the execution phase when the OR endorsement policy is employed, but not when the AND endorsement policy is employed. Soelman et al. ^[26] start from the common needs and requirements of blockchain networks to study the impact of different endorsement policies on the data authenticity of a ledger and give the corresponding improvement measures.

3. EOV Architecture of Hyperledger Fabric

3.1. Key Concepts

As an open-source blockchain platform designed for enterprise users, Hyperledger Fabric provides an exceptional modular and extensible architecture that facilitates its functionality and versatility. Here are several key concepts ^[27] that form the foundation of Hyperledger Fabric, as shown in **Figure 1**. As a whole, a fabric network consists of four modules, called Membership, Blockchain, Transactions and Chaincode, and the four modules are connected by the event stream. For users and developers, they can easily use or develop Hyperledger Fabric using the command line interface (CLI), software development kit (SDK) and application programming interface (API).

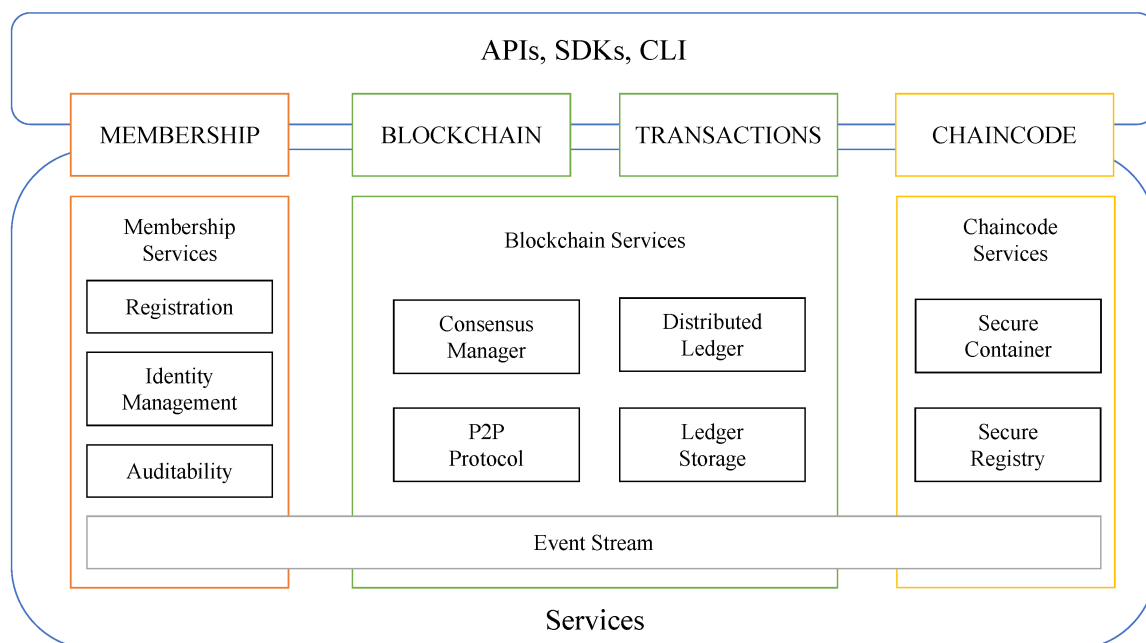


Figure 1. Key components in Fabric network.

- Ledger

Ledger refers to a tamper-resistant, distributed database that stores important factual information concerning business in a blockchain network, including both the current value of the attributes of the objects and the history of transactions that resulted in these current values.

- Node

In Hyperledger Fabric, nodes play a crucial role as communication entities of the blockchain network. They are distributed across the network and collaborate to achieve consensus on the validity of transactions and maintain the shared ledger. In Hyperledger Fabric v2.x, the types of nodes can be broadly classified into two categories, i.e., peer nodes and orderer nodes. In addition, although all peer nodes are identical in Fabric, they can assume multiple roles depending on the network configuration and functional requirements. These roles include Committer, Leader, Endorser and Anchor. **Figure 2** below shows the connection between different nodes [28]. Thus, the transaction process in Fabric can be viewed as a process of interaction among multiple nodes.

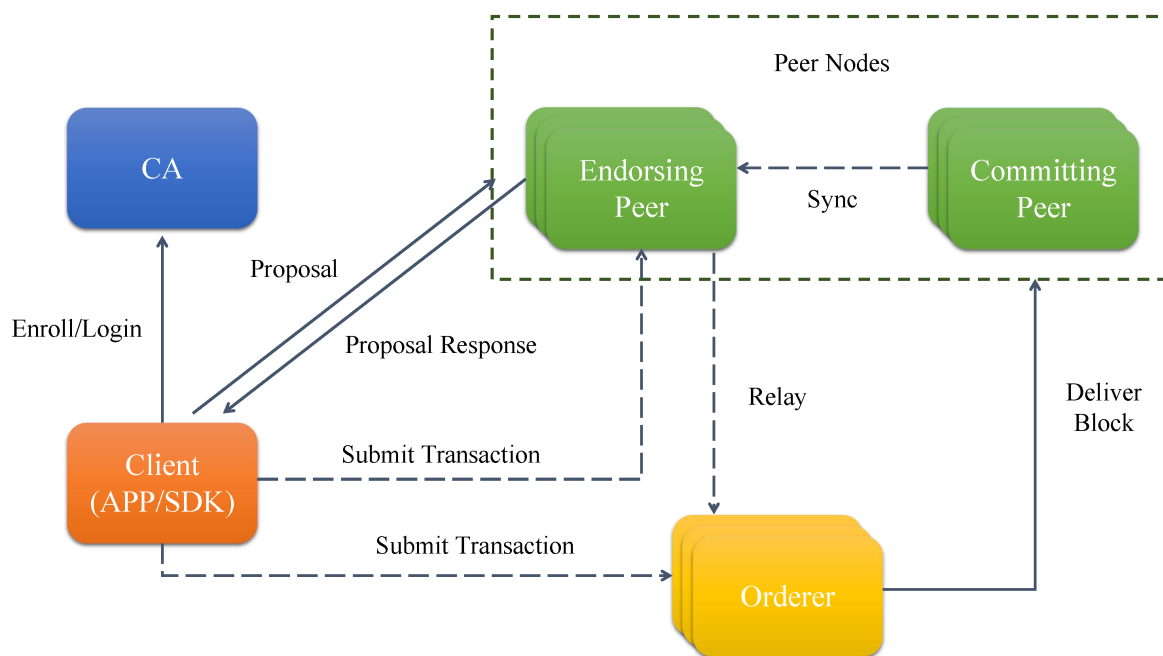


Figure 2. Connection between different nodes.

- Membership Service Providers (MSP)

An MSP is a Hyperledger Fabric component that offers an abstraction of membership operations [29]. In particular, an MSP abstracts away all cryptographic mechanisms and protocols behind issuing certificates, validating certificates, and user authentication. In other words, an MSP manages the identities and access control of participants in the blockchain network. It is responsible for the authentication and authorization of interacting entities within the network like organizations, peers and clients.

- Channel

In a fabric network, there can be multiple channels and each channel is isolated from each other. Participants in the same channel have a separate ledger and maintain it together. In addition, each transaction on the network is executed on a channel, where each party must be authenticated and authorized to transact on the corresponding channel using an MSP.

- Smart Contract/Chaincode

A smart contract is an agreement that automatically executes relevant contract clauses based on business logic set by pre-written code in the context of a blockchain. In Fabric, a smart contract is also called a chaincode. It specifies how data should be validated, processed and updated on the ledger [\[30\]](#)[\[31\]](#).

3.2. Transaction Flow

Transactions flow through the various components of the Fabric network and require the collaborative efforts of the nodes in the network, as illustrated in **Figure 3**. The EOv process starts when the transaction is proposed and ends when the transaction is submitted to the ledger [\[6\]](#). First of all, the client creates a transaction and sends it to a number of endorsing peers. Once receiving the transaction sent by the client, each endorsing peer proceeds to execute the transaction against its own state database, ultimately computing the read–write set specific to that transaction. If the transaction is executed without errors, it signifies that the committed transaction has successfully passed the business rule checks. As a consequence, the endorsing peer will provide its endorsement, signifying its approval to the client. Later, the client waits for a sufficient number of endorsements and then sends the transaction with its endorsement to the orderer nodes that implement the ordering service. The ordering service returns a response to the client after the transaction has been accepted for inclusion in the block. The orderer nodes will reach a consensus on the order of incoming transactions and subsequently create individual blocks in succession from the ordered transactions based on the `BatchSize` and `BatchTimeout` parameters configured by the Fabric network. Once a block is created, the orderer nodes broadcast it to all peers who subsequently proceed to carry out the following procedures. The peers then validate all transactions within the block and commit the block to the ledger and state database once it has passed. In addition, one of the peers will notify the client that the transaction has been committed.

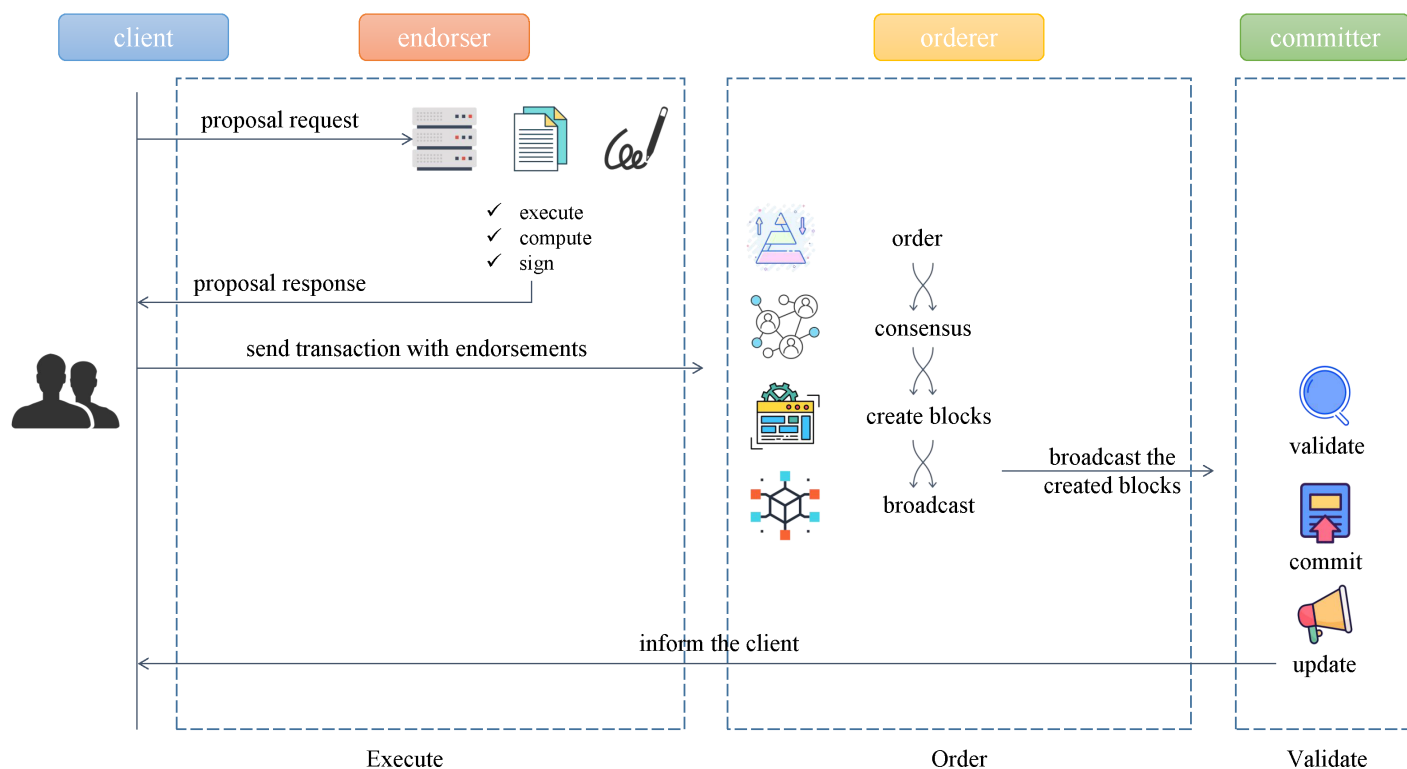


Figure 3. Transaction flow in Fabric.

3.3. Endorsement Flow in Fabric

To begin, once a transaction proposal is generated by the client, the client sends it to the endorsing peers along with its own signature for execution. Subsequently, the client awaits the receipt of a proposal response from the endorsing peers, which includes the simulation results and the endorsement signatures. As for the endorsing peers, when they receive the transaction proposal from the client, they utilize an MSP to validate and authenticate the identity of the client. Later, a chaincode is executed for simulation in an isolated space within the container, which is also called a sandbox. Notice that each endorsing peer executes the same transaction simulation independently and eventually generates an endorsement result for that transaction in the execution phase. Through simulation, the endorsing peers get read–write sets of transactions. The read set is a collection of key-value pairs that the transaction needs to read from the ledger while the write set is a collection of key-value pairs that the transaction intends to write or update to the ledger. Next, the read set will be validated and the write set will be executed. If no error occurs at this point, the endorsing peers will provide the endorsement signatures, attach them to the transaction proposal and send the endorsement signatures back to the client. Once the client collects a sufficient number of endorsement signatures, it proceeds to commit the transaction in the subsequent ordering phase. The pseudocode, as shown in Algorithm 1, of how endorsement is implemented in Hyperledger Fabric's open-source code can help better understand the process.

Algorithm 1: Endorsement phase in Fabric

```

1 begin
2   initialize;
3   check and validate the transaction proposal;
4   if the transaction proposal is valid then
5     execute chaincode for simulation;
6     acquire the read-write set concerning the transaction;
7     validate the read set;
8     execute the write set;
9     if no error then
10      get execution results;
11      perform endorsement signing;
12      send the endorsement results back to the client;
13    end
14  end
15  if the endorsement policy is satisfied then
16    client sends transaction with endorsements to orderer nodes;
17    proceed to the subsequent ordering phase;
18  end
19 end

```

TN,

IoT.

IEEE

Internet

Blockchain. In Proceedings of the 2022 Fourth International Conference on Blockchain

Computing and Applications (BCCA), San Antonio, TX, USA, 5–7 September 2022; pp. 20–26.

6. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
7. Antwi, M.; Adnane, A.; Ahmad, F.; Hussain, R.; ur Rehman, M.H.; Kerrache, C.A. The case of HyperLedger Fabric as a blockchain solution for healthcare applications. *Blockchain Res. Appl.* 2021, 2, 100012.
8. Wang, S.; Taha, A.F.; Wang, J.; Kvaternik, K.; Hahn, A. Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids. *IEEE Trans. Syst. Man Cybern. Syst.* 2019, 49, 1612–1623.
9. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* 2019, 6, 8076–8094.
10. Dabbagh, M.; Choo, K.K.R.; Beheshti, A.; Tahir, M.; Safa, N.S. A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. *Comput. Secur.* 2021, 100, 102078.
11. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276.

12. Kwon, M.; Yu, H. Performance improvement of ordering and endorsement phase in hyperledger fabric. In Proceedings of the 2019 Sixth International Conference on Internet of things: Systems, Management and Security (IOTSMS), Granada, Spain, 22–25 October 2019; pp. 428–432.
13. Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to scalability of blockchain: A survey. *IEEE Access* 2020, 8, 16440–16455.
14. Khan, D.; Jung, L.T.; Hashmani, M.A. Systematic literature review of challenges in blockchain scalability. *Appl. Sci.* 2021, 11, 9372.
15. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Liu, Y. A survey on the scalability of blockchain systems. *IEEE Netw.* 2019, 33, 166–173.
16. Gorenflo, C.; Lee, S.; Golab, L.; Keshav, S. FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. *Int. J. Netw. Manag.* 2020, 30, e2099.
17. Nakaike, T.; Zhang, Q.; Ueda, Y.; Inagaki, T.; Ohara, M. Hyperledger fabric performance characterization and optimization using goleveldb benchmark. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020; pp. 1–9.
18. Berendea, N.; Mercier, H.; Onica, E.; Riviere, E. Fair and efficient gossip in hyperledger fabric. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 190–200.
19. Hang, L.; Kim, B.; Kim, D. A transaction traffic control approach based on fuzzy logic to improve hyperledger fabric performance. *Wirel. Commun. Mob. Comput.* 2022, 2022, 2032165.
20. Sharma, A.; Schuhknecht, F.M.; Agrawal, D.; Dittrich, J. How to databasify a blockchain: The case of hyperledger fabric. *arXiv* 2018, arXiv:1810.13177.
21. Yamashita, K.; Nomura, Y.; Zhou, E.; Pi, B.; Jun, S. Potential risks of hyperledger fabric smart contracts. In Proceedings of the 2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Hangzhou, China, 24 February 2019; pp. 1–10.
22. Liu, C.; Li, M.; Wang, Y.; Wang, Y.; Huo, D.; Chen, Y. Achieve better endorsement balance on blockchain systems. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 581–586.
23. Shammam, E.A.; Zahary, A.T.; Al-Shargabi, A.A. An attribute-based access control model for Internet of things using hyperledger fabric blockchain. *Wirel. Commun. Mob. Comput.* 2022, 2022, 6926408.
24. Qin, X.; Huang, Y.; Yang, Z.; Li, X. LBAC: A lightweight blockchain-based access control scheme for the internet of things. *Inf. Sci.* 2021, 554, 222–235.

25. Wang, C.; Chu, X. Performance characterization and bottleneck analysis of hyperledger fabric. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 1281–1286.
26. Soelman, M.; Andrikopoulos, V.; Pérez, J.A.; Theodosiadis, V.; Goense, K.; Rutjes, A. Hyperledger fabric: Evaluating endorsement policy strategies in supply chains. In Proceedings of the 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Oxford, UK, 3–6 August 2020; pp. 145–152.
27. Hyperledger-Fabricdocs Documentation. Available online: <https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html> (accessed on 16 October 2023).
28. Xu, L.; Chen, W.; Li, Z.; Xu, J.; Liu, A.; Zhao, L. Solutions for concurrency conflict problem on Hyperledger Fabric. *World Wide Web* 2021, 24, 463–482.
29. Ravi, D.; Ramachandran, S.; Vignesh, R.; Falmari, V.R.; Brindha, M. Privacy preserving transparent supply chain management through Hyperledger Fabric. *Blockchain Res. Appl.* 2022, 3, 100072.
30. Foschini, L.; Gavagna, A.; Martuscelli, G.; Montanari, R. Hyperledger fabric blockchain: Chaincode performance analysis. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
31. Rauchs, M.; Glidden, A.; Gordon, B.; Pieters, G.C.; Recanatini, M.; Rostand, F.; Vagneur, K.; Zhang, B.Z. Distributed Ledger Technology Systems: A Conceptual Framework. 2018. Available online: <https://ssrn.com/abstract=3230013> (accessed on 16 October 2023).

Retrieved from <https://encyclopedia.pub/entry/history/show/115890>