Hyperledger

Subjects: Computer Science, Information Systems

Contributor: Zeqi Leng, Kunhao Wang, Yuefeng Zheng, Xiangyu Yin, Tingting Ding

Internet of things (IoT) systems based on blockchain have significant shortcomings in terms of scalability, flexibility, robustness, and privacy. To address these issues, Hyperledger is considered as an ideal technology and attracted a lot of attention. In addition to having the general characteristics of blockchain, Hyperledger achieves new empowerment in four aspects: security, interoperability, consensus, and performance.

Keywords: hyperledger ; blockchain

1. Introduction

Hyperledger is committed to developing an enterprise-grade standard blockchain. At the official conceptual level, Hyperledger is a “greenhouse” system, where all technology is developed by the community. It provides an open source and secure collaborative environment for users, developers, and vendors across all domains. As a result, Hyperledger encourages interoperability between participants in similar domains, each of whom communicates to obtain the necessary information. This effective collaboration greatly reduces the duplication of work for each participant, allowing participants to have more energy to incubate new ideas. To improve code quality, the Technical Steering Committee (TSC) regularly checks the community’s code and projects, and substandard code and projects are discarded. In addition, Hyperledger encourages the achievement of specialization [1], i.e., more people focus on fewer tasks and increase the level of expertise of the participants. Developing the specialization of the participants also helps to promote uniformity of intellectual property rights, and any participant contributing to the Hyperledger community does not have to worry about hidden legal issues.

A generic architecture for Hyperledger should have nine components: a consensus layer, a contract layer, a communication layer, a data storage module, an encryption module, an identity services module, a policy services module, application programming interfaces (APIs), and an interoperability module [2]. These components form a highly modular structure in which the failure of any one component does not affect the overall operation.

As one of the largest open source projects, Hyperledger currently has 18 top projects (including one that was phased out). These top projects provide key technologies for Hyperledger and enable Hyperledger to be widely used in various fields. On the technology side, Hyperledger covers areas such as cross-system authentication, permission control, multi-channel (multi-chain) platform, visualization interface, mobile application, benchmarking, encryption library, Ethernet client, and its business logic development. In terms of applications, Hyperledger is used in the mainstream fields of the internet of things, digital healthcare, supply chain traceability, finance, digital evidence, artificial intelligence, etc.

Hyperledger divides the current top projects into four categories, including distributed ledgers, domain specific, libraries, and tools. Each project contributed to Hyperledger requires regular maintenance by the developers, which means that in addition to maintaining the normal operation of the project, the developers also have to solve problems for members who want to participate in the project in a timely manner. TSC regularly reviews the maintenance status of each project and decides whether the project will move to the next stage. When a project is no longer recommended for use, it will be abandoned by the community after 6 months. However, the abandoned project information and part of the code remain in the community. Each project in Hyperledger must possess the five features of being modular, highly secure, interoperable, cryptocurrencyagnostic, and complete with APIs. Modular components are suitable for developing distributed solutions with different requirements, and high security ensures enterprise-grade blockchain implementation. Interoperability and rich APIs give large enterprise distributed networks easy information interaction.

In general, all projects in Hyperledger go through six phases (status): proposal, incubation, gradated (active), dormant, deprecated, and end of life. The status of each project is dynamic and is jointly determined by the maintainer of the project and the TSC with multiple reviews. At this stage, the top projects in Hyperledger have only two statuses, graduated and incubation. The projects in graduated status are the most active projects with the most members and the most contributed code. Due to constant updates, active projects provide a more mature technology and infrastructure for Hyperledger. Based on the information provided on the official website, the following sections dissected the core architecture and innovative design of the project in graduated status.

2. Fabric

Fabric is the cornerstone of Hyperledger; its innovative design enables Hyperledger to be widely used in various fields [3]. Fabric pioneered the introduction of the authority mechanism, giving the possibility of confidential transactions and ledger isolation in various industries. Fabric's architecture consists of membership services, certificate authorities (CA), nodes, peers, and four types of components. Membership services provide digital certificates for blockchain nodes, CAs provide identity certificates for all nodes in the network, which complete transactions with private and public keys, nodes consist of nodes that are allowed to join the network, and peers are roles that perform different tasks in the blockchain network.

3. Sawtooth

The innovative design of Sawtooth is to simplify the development process of the blockchain application by separating the central system from the application layer [4]. Each Sawtooth node consists of a fixed component validator, and a possible components transaction processor, REST (representational state transfer) API, and client [5]. In the Sawtooth network, the initial node sends broadcast packets to get nearby nodes, and neighboring nodes can join the network according to the rules and broadcast their neighbor's one-hop-away node. As long as there is a response, the node can join the network.

Sawtooth architecture has five core components, including a peer-to-peer network, distributed log, state machine/smart contract logic layer, distributed state storage, and consensus algorithm. The peer-to-peer network allows nodes to communicate via TCP, including information about blocks, peers, etc. [6]. The Sawtooth network broadcasts transactions via gossip protocol. The distributed log includes an ordered list of transactions, which is sorted by nodes according to the consensus algorithm. Sawtooth extends the functionality of smart contracts by treating them as state machines or transaction processors. In the smart contract logic layer, Sawtooth uses radix Merkle. The consensus component provides a consensus interface that allows various consensus algorithms.

4. Iroha

Iroha also provides a distributed framework that is designed to feature privilege management, fault tolerance, and performance efficiency [7]. Compared to other platforms, Iroha requires authorization to read and write data in addition to the authorization required for nodes to join the network. Iroha allows rich built-in commands for simpler asset management, unlike other platforms that require predefined assets [8]. It is designed with a fault-tolerant consensus algorithm, Crash, which allows Iroha to have lower latency.

Iroha architecture has 11 components, including Torii, MST processor, peer communication service (PCS), ordering gate, ordering service, verified proposal creator (VPC), block creator, block consensus (YAC), synchronizer, Ametsuchi blockstore, and world state view (WSV) [9]. In a typical Iroha transaction, client-initiated transactions are received by Torii and forwarded to the MST processor, which typically has two tasks, including forwarding transactions to the PCS and receiving transaction messages (multiple signatures) from other peers. The ordering gate verifies the stateless transactions with other peers, and the ordering service in the peer creates a transaction proposal (each node contains an ordering service) and verifies that the stateless transaction passes the first verification. The VPC performs state verification of the transaction, and the block creator creates new blocks and sends them to the YAC to perform consensus. The YAC forwards the final message to multiple peers. The synchronizer is responsible for downloading blocks from the block store and adding the missing blocks from the peers to the peers. At this point, the Iroha network updates the WSV.

5. Indy

The innovative design of Indy lies in a decentralized identity system [10]. The core feature of this authentication is the self-sovereign identity [11]. This means that once an identity is established, it cannot be revoked, selected, or associated by any institution or person without the permission of the identity owner. There are only two types of nodes in the Indy network, including verification nodes (which are few in number) and observer nodes (which are many in number). Among them, authentication nodes are responsible for processing write requests and participating in consensus. Observer nodes are responsible for reading requests and have the opportunity to become verifying nodes depending on their reputation level. Indy can provide users with portable proof of identity and does not require centralized authentication by a third party.

6. Aries

As the only technology of the six active projects that is not a distributed ledger platform, Aries is a way to provide secure communications for decentralized identity management and verifiable credentials. Aries has four core components, including agents, DID communications, protocols, and key management [12]. Agents provide trusted agents for self-sovereign identity authentication. Specifically, trusted agents help people or organizations send bytes and store data directly. The user downloads or writes the appropriate agent according to the requirements of the agent, such as IoT agents, cloud agents, protocols, scale, and privacy requirements. DID communications is meant to provide information exchange for multiple trusted agents [13]. It is based on decentralized protocols, and its main paradigms are message-based, asynchronous (request–response messages), and simplex. Key management provides a distributed key management system that uses three types of keys, including master keys, key encryption keys, and data keys. The distributed key management system allows any identity owner to perform network connectivity, key exchange, and recovery without relying on any organization, free from the central failures of third-party organizations.

The emergence of Aries facilitates the implementation of decentralized authentication, and peer-to-peer certificate authentication will eradicate the surveillance economy. This authentication method is highly portable and applicable, allowing users to store their proof of employment, or other identification, in a wallet and decide which part of the information can be publicly queried.

7. Besu

Besu is an enterprise class Ethereum platform [14]. Besu has seven core modules, including Ethereum virtual machine (EVM), P2P network, storage, permissioning, privacy, user-facing API, and monitoring.

In terms of privacy, Besu ensures private interactions through Tessera nodes [15]. For example, if a private transaction is sent by Bob, this transaction must first be passed to Bob's Tessera node and complete the information exchange with Alice's Tessera node (the Tessera node involved in the transaction) before being passed to Alice. For better enterprise orientation, permissioning enables node permissions and account permissions so that only specific Storage will store the blockchain and world state, where world state includes account state, account storage, and code storage. Besu provides users with a monitoring interface to demonitor nodes and networks.

Besu supports two node types, including full nodes and archive nodes. Full nodes store only the current block state, ensuring the current up-to-date state. Archive nodes are responsible for storing all the historical states of the blocks since the creation of the world, in addition to the latest state. In addition, Besu provides three APIs for users, including JSON-RPC based on HTTP/WebSockets, RPC publish/subscribe based on WebRocket, and GraphQL based on HTTP.

Besu is compatible with the main Ethernet network and supports both public and private networks. It gives the possibility of building an enterprise class Ethernet platform.

References

1. Blummer, T.; Sean, M.; Cachin, C. An Introduction to Hyperledger; Hyperledger Organization: San Francisco, CA, USA, 2018; Available online: https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf (accessed on 18 December 2021).
2. Leng, Z.; Tan, Z.; Wang, K. Application of Hyperledger in the Hospital Information Systems: A Survey. IEEE Access 2021, 9, 128965–128987.
3. Elrom, E. Hyperledger. In The Blockchain Developer; Apress: Berkeley, CA, USA, 2019; pp. 299–348.
4. Hyperledger, S. Introduction. Available online: https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html (accessed on 17 March 2019).
5. Ampel, B.; Patton, M.; Chen, H. Performance modeling of hyperledger sawtooth blockchain. In Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 1–3 July 2019; pp. 59–61.
6. Moriggl, P.; Asprion, P.M.; Schneider, B. Blockchain technologies towards data privacy—hyperledger sawtooth as unit of analysis. In New Trends in Business Information Systems and Technology; Springer: Cham, Switzerland, 2021; pp. 299–313.

7. Vlachou, V.; Kontzinos, C.; Markaki, O.; Kokkinakos, P.; Karakolis, V.; Psarras, J. Leveraging Hyperledger Iroha for the Issuance and Verification of Higher-Education Certificates. Int. J. Educ. Pedagog. Sci. 2020, 14, 755–763.
8. Iushkevich, N.; Lebedev, A.; Šketa, R.; Takemiya, M. D3ledger: The decentralized digital depository platform for asset management based on hyperledger iroha. In Proceedings of the OTS 2019 Advanced Information Technology and Services, Maribor, Slovenia, 18–19 June 2019; pp. 29–36.
9. Available online: https://github.com/hyperledger/iroha/blob/main/README.md (accessed on 5 January 2022).
10. Dunphy, P. A Note on the Blockchain Trilemma for Decentralized Identity: Learning from Experiments with Hyperledger Indy. arXiv 2022, arXiv:2204.05784.
11. Bhattacharya, M.P.; Zavarsky, P.; Butakov, S. Enhancing the security and privacy of self-sovereign identities on hyperledger indy blockchain. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 16–18 June 2020; pp. 1–7.
12. Available online: https://www.edx.org/course/identity-in-hyperledger-aries-indy-and-ursa (accessed on 5 January 2022).
13. Abramson, W.; Hall, A.J.; Papadopoulos, P.; Pitropakis, N.; Buchanan, W.J. A distributed trust framework for privacy-preserving machine learning. In Proceedings of the International Conference on Trust and Privacy in Digital Business, Bratislava, Slovakia, 3 June 2020; pp. 205–220.
14. Dalla Palma, S.; Pareschi, R.; Zappone, F. What is your distributed (hyper) ledger? In Proceedings of the 2021 IEEE/ACM 4th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Madrid, Spain, 31 May 2021; pp. 27–33.
15. Available online: https://besu.hyperledger.org/en/stable/ (accessed on 6 January 2022).

Retrieved from https://encyclopedia.pub/entry/history/show/63716