information

MDPI

Article

Construction of a Human Resource Sharing System Based on Blockchain Technology

Guoyao Zhu [1], Zhenyu Gu [2,*] and Yonghui Dai [3,*]

[1] Network Security Department, Information Center of Zhejiang Human Resources and Social Security, Hangzhou 310000, China
[2] School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai 201620, China
[3] Management School, Shanghai University of International Business and Economics, Shanghai 201620, China
[*] Correspondence: gzy@suibe.edu.cn (Z.G.); daiyonghui@suibe.edu.cn (Y.D.)

Abstract: Human resource data sharing is very important for the cooperation of human resource institutions. Since the human resource data-sharing service needs to take into account the needs of individuals and talent service institutions, it faces issues such as the security of information sharing and the traceability of information. Therefore, this paper constructs a human resource data-sharing service system based on blockchain technology. Its trust mechanism is based on the Fabric alliance chain, and the system makes full use of its advantages of decentralization and consensus. Our research mainly includes data sharing architecture design, consensus mechanism analysis, smart contract design, data sharing process, and blockchain construction. The contribution of this paper is mainly in two aspects. On the one hand, it explores the trust mechanism of human resource data sharing and gives the scheme of the Fabric alliance chain. On the other hand, the overall architecture and smart contract design are given on the construction of the blockchain, which provides a reference for future research on human resource data sharing.

Keywords: data sharing service; encryption technology; smart contract; consensus mechanism; blockchain technology

check for updates

Citation: Zhu, G.; Gu, Z.; Dai, Y. Construction of a Human Resource Sharing System Based on Blockchain Technology. *Information* **2023**, *14*, 177. https://doi.org/10.3390/info14030177

Academic Editors: Inam Ullah Khan, Mariya Ouaissa, Tarandeep Kaur Bhatia and Syed Bilal Hussain Shah

Received: 29 December 2022
Revised: 27 February 2023
Accepted: 8 March 2023
Published: 11 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the rapid development of information technology has greatly improved the information level of human resource management, and the human resource service system is more and more intelligent. In particular, advanced technologies such as big data compression and electronic storage provide support for the development of human resource image and video data services. However, there are still some problems to be solved in the field of human resource management, among which the sharing of human resource data is one of the most important problems [1]. Human resource data sharing is difficult because it involves cost, trust, security, and privacy protection [2]. For example, personal profile data is often stored in various talent corporations. The storage of these data is discrete, fragmented, and difficult to exchange effectively [3]. In addition, if a talent corporation places its data on a central server, the human archive information will be leaked once the server is successfully hacked. Therefore, some talent corporations use cloud service technology to realize the sharing of talent resume data [4]. They upload electronic archives of information and videos to a cloud service system for data storage and sharing [5]. While the above approach enables data sharing, it also brings some problems, such as the risk of centralized data theft and tampering.

Considering the decentralized, traceable, tamper-resistant characteristics of blockchain technology [6,7], it can well solve the privacy protection and security protection problems of talent archive data. In particular, the integration of blockchain technology in human resource management can well solve some problems in human resource management, such as data sharing and fake diplomas. Therefore, this paper proposes a human resource data-sharing method based on blockchain technology, which provides a new idea for the open cooperation of talent corporations.

2. Related Literature

2.1. Research on Data Sharing for Human Resource Systems

In the research of human resource sharing, many scholars have published research papers. We have summarized the previous scholars' research literature, which is summarized as the research of data sharing technology and application [8,9]. For example, they studied the sharing service model based on cloud computing technology, and proposed the construction and key elements of the service model, the results showed that the model improved the service efficiency and service quality [10]. In order to ensure the security of shared data, a Secure Data Sharing (SDS) framework was proposed. The framework adopts the method of proprietary encryption and proxy re-encryption to solve the problems of authorization and data disclosure [11], Their research provides help for the sharing framework of our study.

In recent years, micro-service architecture and blockchain technology are also gradually applied to data-sharing service systems. Some microservice systems based on the underlying framework of Spring Cloud not only have good security and scalability but also can quickly respond to needs [12]. Therefore, it is used in the construction of digital resource data centers to provide services for various applications and data sharing of governments and enterprises [13,14]. Based on the tamper-proof and traceable characteristics of blockchain data, scholars established a talent credit evaluation system, provided accurate personal credit evaluation capabilities [15], and applied it to the anti-counterfeiting of electronic qualification certificates, which can effectively prevent the tampering and forgery of electronic certificates [16]. In general, the development of information technology provides more and more convenience for human resource data sharing and also provides a solid foundation for our research.

2.2. Research on Trust Mechanism Related to Blockchain

In the trust study of enterprises or individuals, the trust mechanism of data and the confirmation of ownership is one of the main research contents. Due to the existence of the trust, most of the data of each organization is only used internally, and less is used for data sharing. As a result, the phenomenon of information island is relatively common [17]. The lack of safe and effective data flow channels, as well as the ownership of data owners cannot be confirmed and protected, which is the biggest obstacle to data sharing. Since Nakamoto proposed a decentralized point-to-point digital currency system through network consensus in 2008. Blockchain technology based on workload consensus proof mechanism is gradually used to solve the mutual trust between nodes in a distributed system [18,19], which provides a scheme for the research of trust mechanism.

Time stamps and consensus mechanisms are very important in the blockchain. Among them, the timestamp can effectively prevent tampering [20], if the timestamp is added to the digital document, the document can be traced and cannot be forged [21]. The consensus mechanism solves the problem of trust between nodes in a distributed system, which is the foundation of the trust mechanism in the network [22]. For example, the credit evaluation model based on smart contracts and blockchain technology has been proposed, which has played an important role in enterprise credit evaluation [23]. Some scholars have applied the improved PBFT consensus algorithm to the resource-sharing system and given the system framework. Experiments show that the system has good security and performance, and can meet the trust needs of data sharing [24,25]. With the continuous development of blockchain technology, its traceability and security are adopted by more and more service systems.

3. Methodology and Technology

3.1. Encryption Technology in Blockchain

One of the important contents of blockchain technology is cryptography technology, which mainly includes hash algorithms and asymmetric encryption [26]. In essence, a hash algorithm is a method of data change. It can map input text of any length into output text of fixed length through the hash function, and the fixed length text value is the hash value. The hash algorithm uses a hash function and compression mapping method to transform data. It transforms a long input text into a short hash value, which has the characteristics of irreversibility and uniqueness. Among them, irreversibility means that when the input text is known, the output can be calculated by the hash algorithm, but the input text cannot be pushed out from the output; uniqueness means that if the input text is not the same, then the output must be different, that is, there is no case that two different inputs correspond to the same output value. The hash algorithm is widely used, and digital signature is one of its common applications [27]. The file loaded by the hash function will have a series of hash values. If the file is modified by an attacker, its hash value must be different from the original hash value, which shows that it is a file that has been attacked. There are many hash algorithms, such as MD5, SHA-1, SHA-2, and ShA-3, and SHA-256 is the main hash algorithm used by blockchain [28,29].

Merkle tree is a data structure based on a hash algorithm. It is a binary tree, which is composed of multiple groups of nodes [30]. Each node is the hash of its two child nodes. In the Merkle tree, the data will be divided into several small data blocks and placed at the bottom. The hash value is conducted upward and combined in pairs to form the root node. The root node is formed by the hash of its two child nodes, which represents the "top" of the tree. The formation process of the hash value of the root node of the tree is that the lowest data block first executes the hash algorithm to obtain a hash value, then the hash values of the two adjacent child nodes are merged into one value, and the hash operation is continued. The hash value of a node is obtained, and then this value is merged with the hash value of the adjacent node to generate a new value, and then iteratively repeats the above steps, and finally, all nodes are merged to form a root hash [31]. Therefore, the Merkle tree is often used to judge and check whether the data in the blockchain has been tampered with. The Merkle tree in blockchain connection is shown in Figure 1.

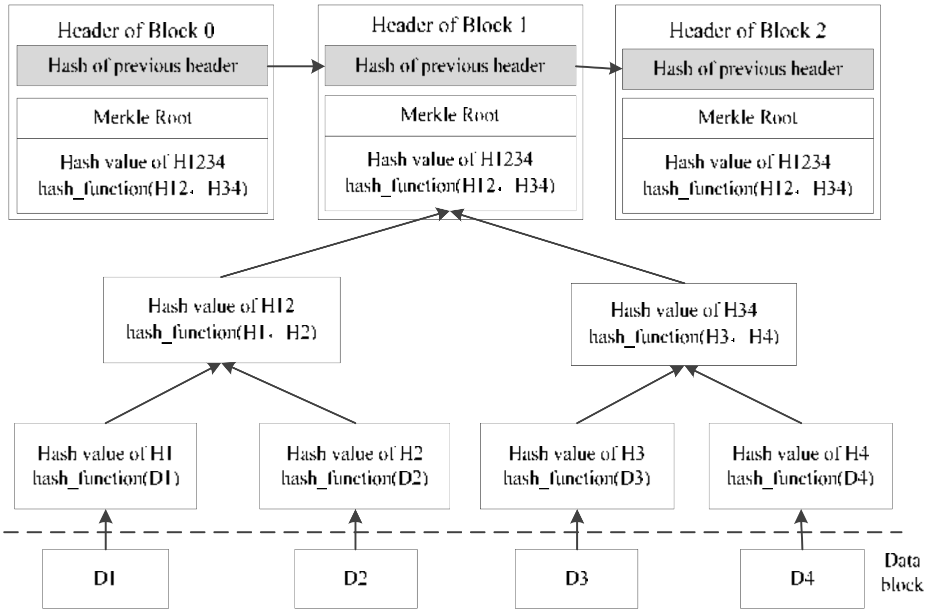

Figure 1. Sample of Merkle tree in blockchain connection.

Another technology of blockchain encryption is asymmetric encryption, which means using different keys for encryption and decryption. Its key is divided into public key and private key [32]. The public key is open to the whole blockchain network and can be freely obtained by all network users. The private key is private to the user and strictly confidential. The asymmetric encryption mechanism makes it impossible for attackers to infer the private key of decryption through the public key, thus ensuring the security of data on the network. Compared with a symmetric encryption algorithm, it has higher security and avoids the defects of key distribution and management. Its advantages are that it can be used for authentication and public key encryption, but the disadvantage is that public key encryption and decryption take a long time. There are many kinds of asymmetric encryption technology, such as RSA (Rivest-Shamir-Adleman), ECDSA (Elliptic Curve Digital Signature Algorithm), ECC (Elliptic Curve Cryptography), CP-ABE (Ciphertext Policy Attribute Based Encryption) algorithm [33,34]. Among these, the CP-ABE algorithm is widely used in the alliance chain of blockchain. In this algorithm, the user's private key corresponds to an attribute set, and the ciphertext corresponds to an attribute policy [35,36]. Only when the attribute set meets the attribute policy, the user can decrypt. It mainly consists of the following steps.

Step 1: Initialization parameters. This step is to initialize the parameters, which includes inputting a random parameter n and executing the initialization algorithm, then getting the Public Key (PK) and Master Cipher Key (MCK). The initialization algorithm is shown as (1).

$$\text{Parameters.Setup(n)} \rightarrow \text{(PK.MCK)} \tag{1}$$

Step 2: Generate Cipher Key. This step is to generate the User Secret Key (USK) according to the previous PK, MCK, and the Set of User-Defined Attribute (Set_UDA); the expression is shown as (2).

$$\text{KeyGen(PK, MCK, Set\_UDA)} \rightarrow \text{USK} \tag{2}$$

Step 3: Data encryption. This step completes the encryption of the data owner. The data owner inputs PK, message (m) to be encrypted, and Decryption Policy (DP), then outputs the ciphertext (c); the expression is shown as (3).

$$\text{DataEncrypt(PK, m, DP)} \rightarrow \text{c} \tag{3}$$

Step 4: Data decryption. This step completes the decryption of the data user. The data user inputs PK, USK, and c, then outputs m; the expression is shown as (4).

$$\text{DataDecrypt(PK, c, USK)} \rightarrow \text{m} \tag{4}$$

3.2. Data Sharing Architecture Based on Fabric Alliance Chain

As a data trust solution, the alliance chain has the characteristics of partial decentralization, internal access to data, and relatively fast transaction, so it is used in our human resource sharing research. Specifically, the architecture based on the Fabric alliance chain is shown in Figure 2.

It can be seen from Figure 2 that the architecture is divided into three layers, that is the data storage layer, the data processing layer, and the application layer. The data storage layer is responsible for the data storage of the alliance chain and the local database, which includes the smart contract database, user database, record, the log chain database, and local database of each node. The data processing layer is responsible for the technical processing of data services, which is based on the Fabric alliance chain. The application layer provides users with various business functions related to human resource data sharing through API (Application Programming Interface) and events. The data processing layer can be seen as having two parts. One part includes member management, consensus service, chaincode service, security, and password service to jointly support the upper business. The other part is mainly connected with identity management, ledger management, transaction management, and smart contract management through the API interface.

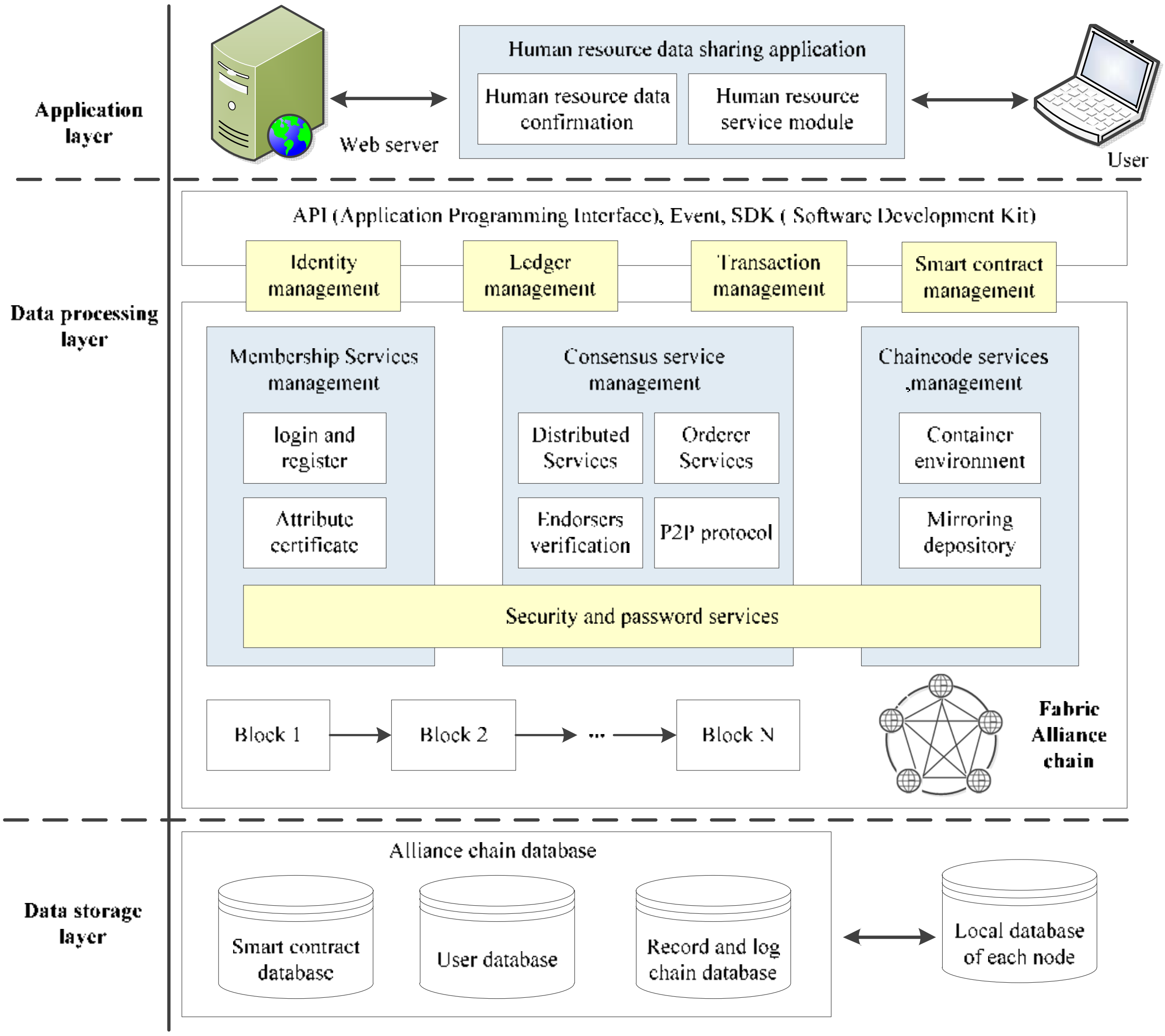

Figure 2. Data sharing architecture based on Fabric alliance chain.

Membership services provider. It is responsible for providing all member management services on the chain, including member registration and certificate management [37]. Users need to authenticate and authorize to become members of the alliance chain first, and then operate and maintain the data of the alliance chain as members to share the data.

Consensus service management. Fabric consensus includes endorsement, sorting, verification, and commitment functions. It is responsible for ensuring the uniqueness and correctness of transaction blocks before they are written into the ledger. It is necessary to ensure that the transaction order of all participants on the chain is unique and the data is synchronized.

Chaincode service management. It is a program running in the Docker container, and its role is similar to that of the smart contract in Ethereum. It is deployed in the network nodes of the fabric alliance chain and completes the communication of corresponding nodes through specific protocols. The most important thing is to use it to operate the data in the distributed ledger.

Security and password service. The Fabric alliance chain provides security and cryptographic services through the special component interface BCCSP (BlockChain Cryptographic Service Provider), which is responsible for encryption and decryption, key pair generation, hash algorithm operation, signature, and verification.

Identity management. There are many types of participants in Fabric, and their identification is realized by calling specific interfaces. The node in the federated network calls the interface and obtains the public key and private key of the user after applying for the user registration certificate. To execute a transaction, the sender must first send the private key signed by the opposite party. After the receiver completes signature verification, the two parties can perform subsequent transaction operations.

Ledger management. After the node members are authenticated and authorized, they can query the ledger through this interface. There are many ways to query, such as querying blocks according to the block hash value, querying transactions according to the transaction number, querying blockchain information according to the channel name, and custom query.

Transaction management. Fabric's ledger data operation is completed through the transaction management interface. When it operates the ledger data, it first sends the transaction information to the endorsement node through a special interface, and then the endorsement node implements the endorsement. After the above steps are completed, it is submitted to the sorting node for packaging to form a block. Finally, the block is synchronized and distributed to each ledger node in the alliance chain network.

Smart contract management. A smart contract in Fabric is called chaincode, which describes contract terms, transaction conditions, and transaction business logic through computer language [38]. It is the implementation carrier for the interaction between the application business and the underlying. After the chaincode is written and instantiated, it can be called by the node to automatically complete the transaction and maintain the ledger data.

3.3. Fabric Consensus Mechanism

The consensus mechanism means that in a distributed system, all nodes share the same view on data or state through information interaction through interfaces, to realize data or state synchronization in a remote environment. As an important part of the blockchain system, the consensus mechanism is responsible for solving the consistency of accounting in the distributed working environment. Because the blockchain network is decentralized based on P2P network, it is different from traditional network in data consistency of each node. Some consensus mechanisms have been proposed, such as PoW (Proof of Work), PoS (Proof of Stake), and DPoS (Delegated Proof of Stake) [39,40]. In the above consensus mechanism, PoW is completely decentralized, which requires high hardware requirements [41]. PoS is a highly decentralized stack and completely decentralized [42], but it has general hardware requirements, and the rights and interests can adopt the coin days. Under this mechanism, if a node has more coin days, the probability that the node generates blocks will be higher. Compared with the PoW mechanism, PoS can save more computing power, but it also has some risks such as the whole network being easy to be controlled by nodes with early coin days. The comparison of the typical consensus algorithm is shown in Table 1.

Table 1. The comparison of the typical consensus algorithm.

Algorithm Name	Degree of Decentralization	Network Type	Hardware Requirements	Technology Maturity	Maximum Fault Tolerance
PoW	Completely decentralized	Public chain	high	mature	>1/2
PoS	Completely decentralized	Public chain	normal	mature	>1/2
DPoS	Completely decentralized	Public chain, Alliance chain	normal	mature	>1/2
PoET	Local decentralization	Alliance chain	high	normal	>1/2
PBFT	Local decentralization	Alliance chain	normal	mature	>1/3

DPoS, PoET (Proof of Elapsed Time), and PBFT (Practical Byzantine Fault Tolerance) are more suitable for the consensus of the alliance chain. Among them, the PoET algorithm is difficult to adopt on a large scale because it uses specific hardware. PBFT algorithm can prevent the bifurcation phenomenon, its advantage is that it can make all nodes consistent, but the disadvantage is that it needs a lot of reliable network communication. DPOS algorithm is developed on the basis of PoS, and it deals with the choice of bookkeeper in a special way. It is completely decentralized and each node has the right to vote. The working principle of DPoS is as follows: First, the recorder is selected based on the number of votes, and then the recorder is voted so that the recorder can verify and keep accounts in turn. In this way, 90% of bookkeepers can go online. Through the above consensus mechanism, the number of nodes participating in verification and accounting in the network can be greatly reduced, to improve the efficiency of reaching consensus, and effectively avoid the high cost of PoW and the high concentration of PoS. In our research, the human resource sharing system is based on the architecture of the Fabric consensus mechanism, and it is a consensus model composed of endorsement, sorting, and verification, which is shown in Figure 3.

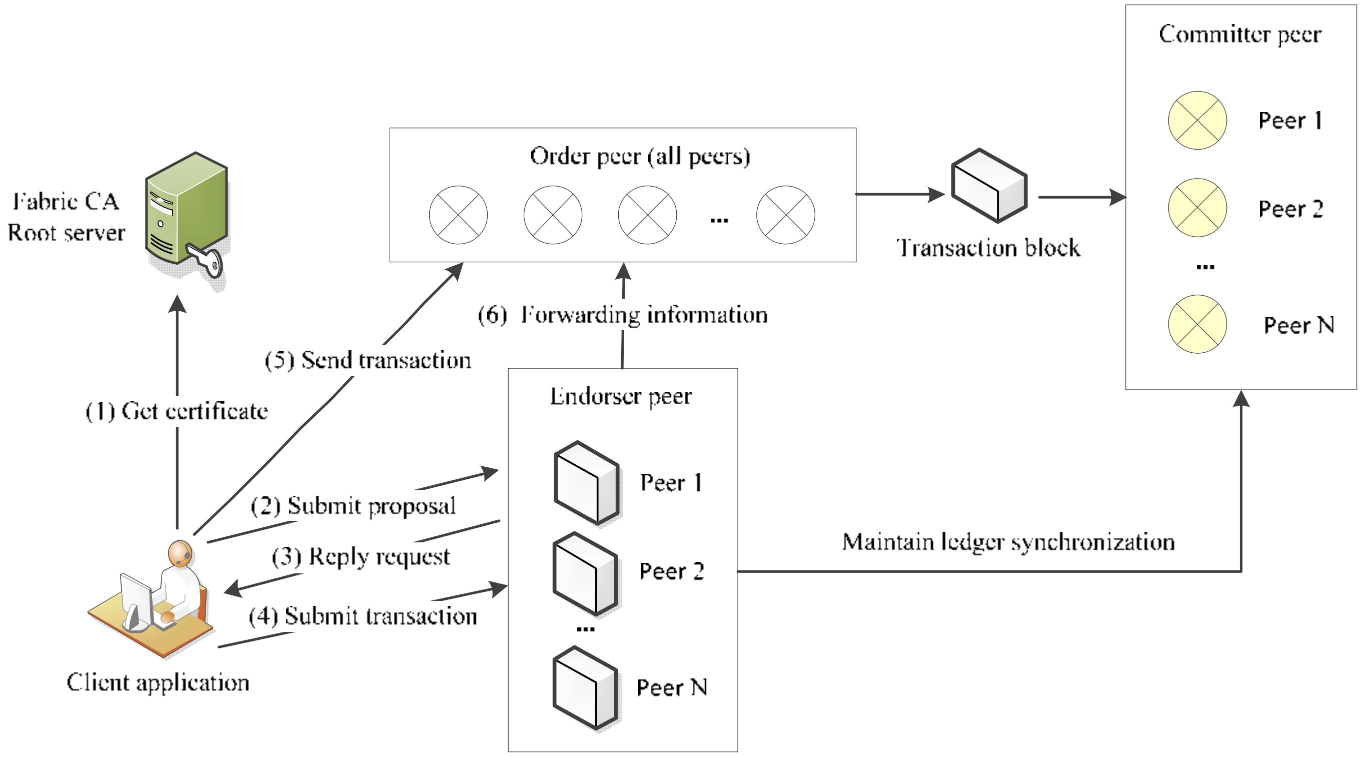

Figure 3. Fabric consensus mechanism.

The Fabric consensus mechanism mainly includes CA Root Server, order peer, endorser peer, and committer peer. First, the client obtains authorization from the CA Root server. Second, the client application submits a proposal to the endorser peer. After the client creates a transaction proposal, it encapsulates the transaction proposal and signature as an endorsement application according to the corresponding endorsement policy. Then it sends them to all endorser peers. Among, proposal mainly includes channel information, chaincode information, timestamp, client signature, etc. Third, endorser peer reply request. Fourth, the client submits a transaction to the endorser peer. Fifth, the client sends a transaction to an order peer. Sixth, the endorser peer forwards information to the order peer. Finally, the order peer is packed into blocks and sent to the committer peer.

4. Design and Implementation of Data Sharing Mechanism

4.1. Design of Data Sharing Process

In the construction of human resource information sharing, we made full use of the advantages of the alliance chain to ensure security, privacy, and storage efficiency in the data-sharing process. There are four main types of roles involved in data sharing, which include HR (Human resources) data owner, HR data requester, HR alliance chain, and Alliance chain data repository. It can be seen from Figure 4 that the HR data owner and HR data requester need to complete registration on the alliance chain. The sequence diagram of the data-sharing process is shown in Figure 4.

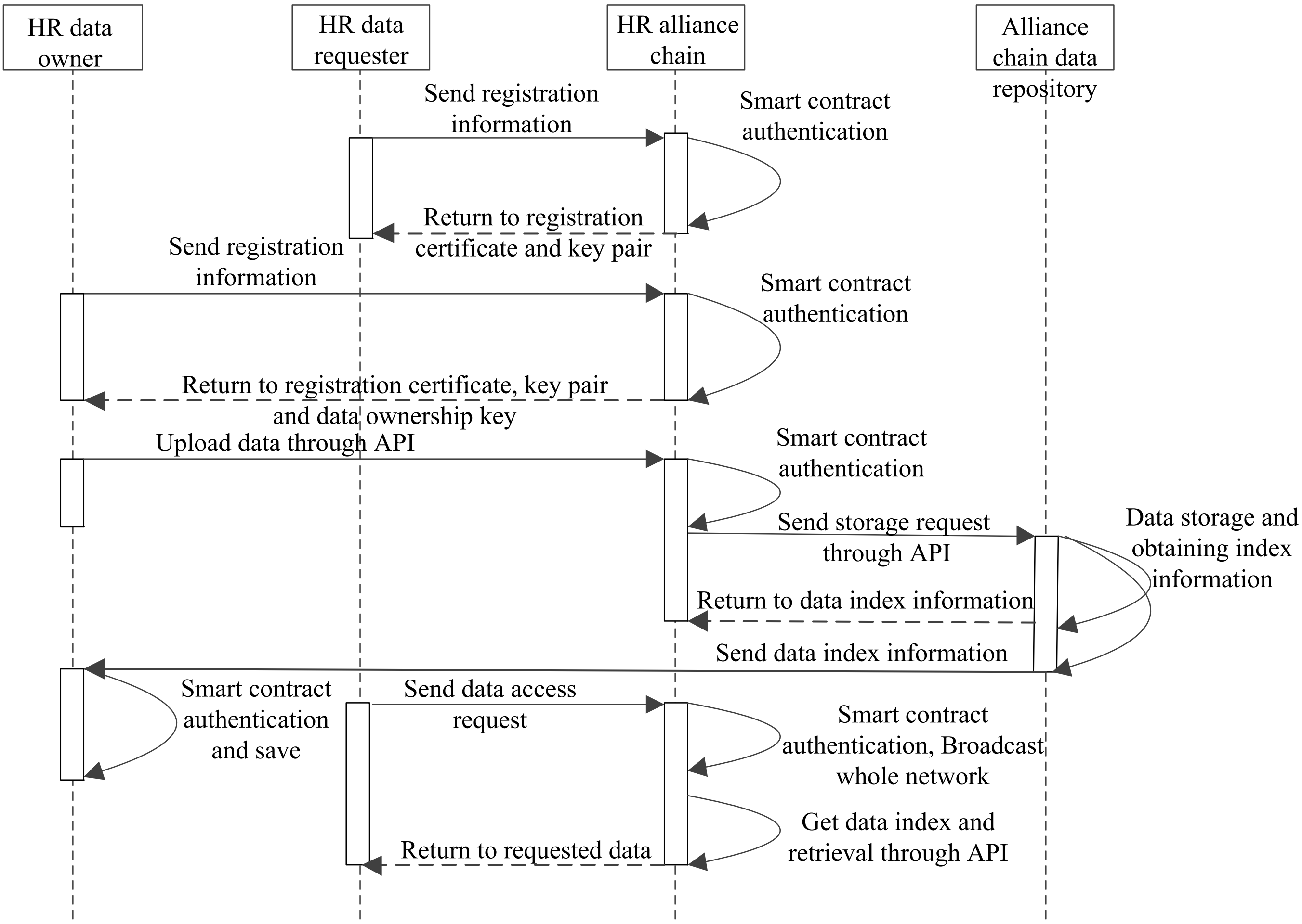
HR data owner
HR data requester
HR alliance chain
Alliance chain data repository
Send registration information
Smart contract authentication
Return to registration certificate and key pair
Send registration information
Smart contract authentication
Return to registration certificate, key pair and data ownership key
Upload data through API
Smart contract authentication
Send storage request through API
Data storage and obtaining index information
Return to data index information
Send data index information
Smart contract authentication and save
Send data access request
Smart contract authentication, Broadcast whole network
Get data index and retrieval through API
Return to requested data

Figure 4. The sequence diagram of the data sharing process.

Step 1: HR data owner and HR data requester send registration information to the HR alliance chain. If their information passes the smart contract verification, the alliance chain will return their registration certificate and key pair. It should be noted that the difference between the returned information received by the two roles is that the data owner will also receive a data ownership key, which can be used to determine the identity of the data.

Step 2: The HR alliance chain receives the registration request, and verifies it according to the existing smart contract. After verification, the results are returned.

Step 3: HR data owner updates data to the HR alliance chain through API. HR alliance chain receives data and verifies it according to the existing smart contract. After verification, it sends a storage request to the alliance chain repository through API.

Step 4: The alliance chain repository completes the data storage and informs the HR alliance chain of the index information of the data. At the same time, it also informs the data owner of the index information of the data. The data owner verifies the existing smart contract and saves it after verification.

Step 5: HR data requester sends a data access request to the HR alliance chain. If the request passes the verification type of data query of the smart contract such as the command of 'peer chaincode query -C -n -c'. The HR alliance chain will broadcast the whole network, and then get data index and retrieval through API. Finally, it returns the results to the data requester through the API.

4.2. Design of Smart Contract

The smart contract of the blockchain is essentially a deterministic program code that can run on the blockchain network node. It carries the predetermined business logic, including the status of objects, the conditions for transaction triggering, and the rules for data updating. If data on the blockchain needs to be maintained and operated, it can only be carried out through smart contracts. The execution of smart contracts needs to have good independence, so it is often carried out in some specific environment, such as a Docker container, or EVM virtual machine. The above execution environment can well ensure that it is executed without interference. In Fabric, the chaincode is called a smart contract. Every chaincode program executes business through interfaces, so if its business is complex, it may have many interfaces. Among them, Init and Invoke interfaces are indispensable in each chaincode program. In this research, there are mainly three kinds of smart contract function modules, which is the user smart contract module, HR data service smart contract module, and contract management module. The following is the design of user smart contracts.

User smart contracts include two smart contracts: user registration and user login smart contracts. It is to provide registration and login services for data owners such as human resources institutions or talents. The user login smart contract is to convert the identity of the login user into the address ID on the blockchain network. At the same time, the address ID is mapped with the login user on the blockchain network. When data owners have network identification addresses, they can perform data-sharing-related operations. When users login to the alliance network, they will pass in parameters. The user login smart contract verifies the parameters and determines whether they meet the preset trigger conditions for logging into the network. At the same time, the preset response rules of the contract also start to match the results and give the corresponding response results. The sample of preset trigger conditions and response rules for users to login to a smart contract is shown as (5) and (6).

$$\text{SC\_loginC} = \{\text{L\_Condition\_1}, \text{L\_Condition\_2}, \cdots, \text{L\_Condition}\} \tag{5}$$

$$\text{SC\_loginR} = \{\text{L\_Rule\_1}, \text{L\_Rule\_2}, \cdots, \text{L\_Rule\_N}\} \tag{6}$$

In (5), SC_loginC is the set of trigger conditions, and L_Condition_N is the specific conditions. For example, one condition is to determine whether the user's name is the same as the user's name parameter during login.

In (6), SC_loginR is the set of response rules, and L_Rule_N is the specific response rule. For example, one response rule is User login succeeded.

4.3. Design of Blockchain Construction

In our system, the human resources information blockchain is composed of a series of blocks with a version number. After the human resources information with block information is generated, the system will connect this information according to the generated time sequence to form the human resources information blockchain. On the blockchain, the blockheads and block data subjects of all blocks are on the chain. The sample of human resources of the storage blockchain is shown in Figure 5.

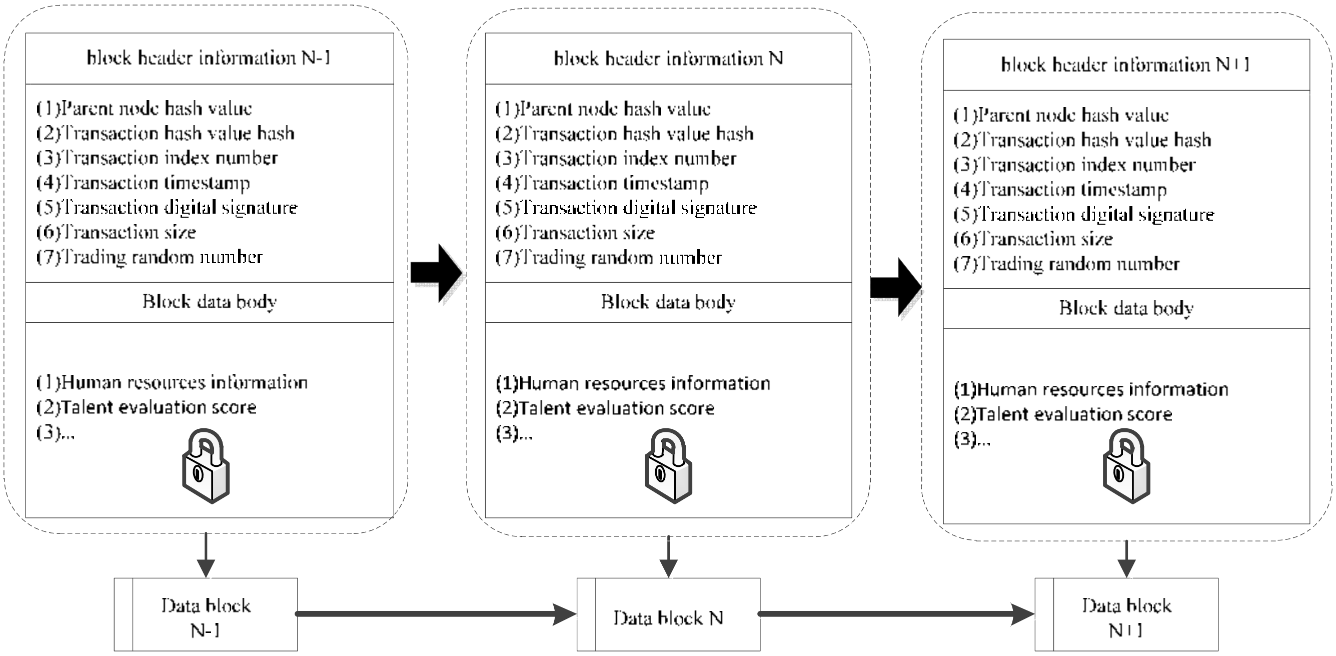
block header information N-1
(1)Parent node hash value
(2)Transaction hash value hash
(3)Transaction index number
(4)Transaction timestamp
(5)Transaction digital signature
(6)Transaction size
(7)Trading random number
Block data body
(1)Human resources information
(2)Talent evaluation score
(3)...
block header information N
block header information N+1
Data block N-1
Data block N
Data block N+1

Figure 5. The sample of construction of blockchain.

It can be seen from Figure 5 that the information on the block header and block data subject mainly includes: the hash value of the parent node of the current block, transaction digital signature of the current block, transaction hash value of the current block, transaction timestamp of the current block, random number of transactions of the current block, human resources information of current block and talent evaluation score of current block Information, etc. In the distributed storage system structure, once the accounting node has a new change, it will broadcast the new record block in the whole network, each node can store it, so each node can make a full or partial backup of data according to its own needs.

4.4. Deployment and Execution of Chaincode

In the implementation of the system, the VUE framework is used for front-end development, and the Go language is used for writing chaincode. Among, the development of the chaincode is completed on the basis of the Fabric blockchain environment. Its deployment and invocation include the following steps.

Step 1: Installation of the runtime environment, which includes Git, Docker, Docker-compose, Node.js, Go language environment, and Fabric system files.

Step 2: Generate Docker Image. The bootstrap.sh script in the Fabric system file needs to be executed, and the docker image and related files will be generated.

Step 3: Deploy the hyperledger fabric. The configuration file of the created node is executed, and the certificate, data file, and Genesis block are generated. Then the order peer node is generated, the channel is created and the node is added to the channel, and the hyperledger fabric deployment is completed. For example, the terminal executes the command to create the Genesis block as follows.

- configtxgen -profile ExamGenesis -channelID byfn-sys-channel -outputBlock./channel-artifacts/genesis.block

After the above command is executed, the genesis.block file is generated under the channel-artifacts directory, which indicates that the creation of the Genesis block is successful.

Step 4: Initialization and use of chaincode. After the chaincode is written in Go language, it needs to be installed to the peer node and instantiated to the channel, and then it can be used after initialization. Specifically, the chaincode is executed by calling the interface in the Invoke method. Example commands for instantiating chaincode and invoking chaincode at the terminal are as follows.

1. peer chaincode instantiate -n testcc -v 1.0 -c '{"Args":["init","role","administrators"]}' -C $CHANNEL_NAME
2. peer chaincode query -C $CHANNEL_NAME -n testcc -c '{"Args":["query","role"]}'

The first command above is executed, and a key-value pair <role, administrators> is initialized and saved to the variable $CHANNEL_NAME channel.

The second command above is executed, and the data corresponding to a key ("role ") will be queried.

In addition, to analyze the performance of the system, we used the Caliper test tool to test the invoke query operation. The test items include send rate, average latency, and throughput. The test result is 207.1 tps, 1.83 s, and 206.7 tps. The results show that it can meet the needs of users.

5. Conclusions

Human resource data sharing is a very important issue for talent service institutions and cooperation between institutions. When HR data is shared, the system should not only consider the needs of talent institutions and enterprises but also consider the security and access convenience of the system. Because blockchain is decentralized, traceable, and tamper-proof, it is a new idea to share human resource data based on blockchain technology. This paper has conducted exploratory research on human resource data sharing, which includes data sharing architecture, consensus mechanism design, smart contract design, data sharing process, and blockchain construction. The practice shows that the above research can well solve the problem of data sharing in different human resource service organizations.

Overall, there are two main conclusions of the study. On the one hand, the research on human resource data sharing based on blockchain technology is very meaningful research, which can help human resource institutions and talents to exchange information more effectively and can effectively ensure the security and traceability of data. On the other hand, the consensus mechanism and partial decentralization of the Fabric alliance chain can well meet the internal access needs of alliance members in human resource sharing. In the future, more human resource service organizations and enterprises can be invited to join the alliance chain to better realize human resource data sharing.

Author Contributions: Conceptualization, G.Z., Z.G. and Y.D.; methodology, G.Z. and Z.G.; software, G.Z., Z.G. and Y.D.; validation, G.Z. and Z.G.; formal analysis, Y.D.; investigation, G.Z. and Z.G.; resources, G.Z. and Y.D.; data curation, Z.G.; writing—original draft preparation, G.Z. and Z.G.; writing—review and editing, G.Z., Z.G. and Y.D.; visualization, Y.D.; supervision, G.Z.; project administration, G.Z.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Major Project of Philosophy and Social Science Research, Ministry of Education of China, grant number 19JZD010, and Project of National Natural Science Foundation of China, grant number 71971066, 72274119.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are currently not publicly available due to participant privacy, but they are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, H.; Chen, B.; Liu, Y.X. Research on personnel file management system based on blockchain. *Comput. Sci.* **2021**, *48*, 713–718.

2. Wang, X.; Feng, L.B.; Zhang, H.; Lyu, J.; Wang, L.; You, Y. Human Resource Information Management Model Based on Blockchain Technology. In Proceedings of the 2017 Conference IEEE Symposium on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 6–9 April 2017.
3. Lai, J. The Application Prospects of Blockchain Technology in Human Resource Management. *Mod. Manag. Forum* **2020**, *4*, 167–171. [CrossRef]
4. Navimipour, N.J.; Navin, A.H.; Rahmani, A.M.; Hosseinzadeh, M. Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources. *Comput. Ind.* **2015**, *68*, 65–77. [CrossRef]
5. Kim, H.W.; Park, J.H.; Jeong, Y.S. Human-centric storage resource mechanism for big data on cloud service architecture. *J. Supercomput.* **2016**, *72*, 2437–2452. [CrossRef]
6. Akbar, M.A.; Victor, L.; Saima, R.; Syed, F.Q.; Sajjad, M.; Ahmed, A. Towards roadmap to implement blockchain in healthcare systems based on a maturity model. *J. Softw. Evol. Process* **2022**, *34*, e2500. [CrossRef]
7. Manikandan, S.; Rahaman, M.; Song, Y.L. Active Authentication Protocol for IoV Environment with Distributed Servers. *Comput. Mater. Contin.* **2022**, *12*, 5789–5808. [CrossRef]
8. Navimipour, N.J.; Rahmani, A.M.; Navin, A.H.; Hosseinzadeh, M. Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources. *Comput. Hum. Behav.* **2015**, *46*, 57–74. [CrossRef]
9. Maqueira Marín, J.M.; Oliveira-Dias, D.D.; Navimipour, N.J.; Gardas, B.; Unal, M. Cloud computing and human resource management: Systematic literature review and future research agenda. *Kybernetes* **2022**, *51*, 2172–2191. [CrossRef]
10. Liu, Z.; Chen, L.F. Research of the service model based on Internet in Human resources sharing service center. *Hum. Resour. Dev. China* **2017**, *7*, 92–98.
11. Samanthula, B.K.; Elmehdwi, Y.; Howser, G.; Madria, S. A secure data sharing and query processing framework via federation of cloud computing. *Inf. Syst.* **2015**, *48*, 196–212. [CrossRef]
12. Li, Y.; Wang, C.Z.; Li, Y.C.; Su, J. Granularity Decision of Microservice Splitting in View of Maintainability and Its Innovation Effect in Government Data Sharing. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 1057902. [CrossRef]
13. Puspitasari, N.; Budiman, E.; Sulaiman, Y.N.; Firdaus, M.B. Microservice API Implementation for E-Government Service Interoperability. In Proceedings of the 2019 International Conference of Science and Information Technology in Smart Administration (ICSINTeSA), Balikpapan, Indonesia, 16–17 October 2019.
14. Huang, N.N.; Shen, L.; Yang, Y.Y. Research on Efficient and Secure Data Sharing Scheme in Personal Health Record on Cloud. *Comput. Eng. Appl.* **2020**, *56*, 92–97.
15. Li, G.L.; Yan, S.; Yuan, L.Z.; Zhang, H.B. System Design of Archives Management Based on Blockchain Technology. In Proceedings of the International Conference on Communications, Information System and Software Engineering (CISSE 2020), Guangzhou, China, 18–20 December 2020.
16. Turkanovic, M.; Podgorelec, B. Signing Blockchain Transactions Using Qualified Certificates. *IEEE Internet Comput.* **2020**, *24*, 37–43. [CrossRef]
17. Hu, N.H.; Gu, W.C.; Zhou, Y.Z. Information sharing, credit availability and credit default risk—Based on the analysis of EU countries' experience from 2004 to 2014. *Shanghai Financ.* **2016**, *6*, 33–39.
18. Mu, Y.; Margheri, A.; Hu, R.; Sassone, V. Differentially Private Data Sharing in a Cloud Federation with Blockchain. *IEEE Cloud Comput.* **2017**, *5*, 69–79.
19. Akbar, M.A.; Mahmood, S.; Siemon, D. Toward Effective and Efficient DevOps using Blockchain. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering 2022, Gothenburg, Sweden, 13–15 June 2022; pp. 421–427.
20. Oyelude, A.A. Trending issues in advancing blockchain technology in libraries, archives and museums. *Libr. Hi Tech News* **2022**, *39*, 6–7. [CrossRef]
21. Kim, Y.; Raman, R.K.; Kim, Y.S.; Varshney, L.R.; Shanbhag, N.R. Efficient Local Secret Sharing for Distributed Blockchain Systems. *IEEE Commun. Lett.* **2019**, *23*, 282–285. [CrossRef]
22. Hu, Y.; Chen, L. Research on data sharing scheme of production line based on blockchain. *Foreign Electron. Meas. Technol.* **2019**, *38*, 123–127.
23. Wang, W. A SME Credit Evaluation System Based on Blockchain. In Proceedings of the 2020 International Conference on E-Commerce and Internet Technology (ECIT), Zhangjiajie, China, 22–24 April 2020.
24. Xu, R.Z.; Zhang, L.; Zhao, H.W.; Peng, Y. Design of Network Media's Digital Rights Management Scheme Based on Blockchain Technology. In Proceedings of the 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), Bangkok, Thailand, 22–24 March 2017.
25. Dai, Y.H.; Li, G.W.; Xu, B. Study on learning resource authentication in MOOCs based on blockchain. *Int. J. Comput. Sci. Eng.* **2019**, *18*, 314–320. [CrossRef]
26. Govindasamy, C.; Antonidoss, A. Enhanced Inventory Management Using Blockchain Technology Under Cloud Sector Enabled by Hybrid Multi-Verse with Whale Optimization Algorithm. *Int. J. Inf. Technol. Decis. Mak.* **2022**, *21*, 577–614. [CrossRef]
27. Alzubi, J.A. Blockchain-based Lamport Merkle Digital Signature: Authentication tool in IoT healthcare. *Comput. Commun.* **2021**, *170*, 200–208. [CrossRef]
28. Amy, M. AI researchers embrace Bitcoin technology to share medical data. *Nature* **2018**, *555*, 293–294.

29. Fu, J.H.; Qiao, S.H.; Huang, Y.Z.; Si, X.M.; Li, B.; Yuan, C. A Study on the Optimization of Blockchain Hashing Algorithm Based on PRCA. *Secur. Commun. Netw.* **2020**, *8*, 8876317. [CrossRef]
30. Tohidi, H.; Vakili, V.T. Lightweight Authentication Scheme for Smart Grid Using Merkle Hash Tree and Lossless Compression Hybrid Method. *IET Commun.* **2018**, *12*, 2478–2484. [CrossRef]
31. Lee, D.; Park, N. Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree. *Multimed. Tools Appl.* **2021**, *80*, 34517–34534. [CrossRef]
32. Zhao, H. A Cross-Border E-Commerce Approach Based on Blockchain Technology. *Mob. Inf. Syst.* **2021**, *2021*, 2006082.
33. Athena, J.; Sumathy, V.; Kumar, K. An identity attribute–based encryption using elliptic curve digital signature for patient health record maintenance. *Int. J. Commun. Syst.* **2017**, *31*, e3439. [CrossRef]
34. Anissa, S.; Medien, Z.; Chiraz, M.; Mohsen, M. Design and Implementation of Low Area/Power Elliptic Curve Digital Signature Hardware Core. *Electronics* **2017**, *6*, 46.
35. Liu, Z.; Wang, F.; Chen, K.; Tang, F. A New User Revocable Ciphertext-Policy Attribute-Based Encryption with Ciphertext Update. *Secur. Commun. Netw.* **2020**, *6*, 8856592. [CrossRef]
36. Gao, S.; Piao, G.; Zhu, J.; Ma, X.D.; Ma, J.F. TrustAccess: A Trustworthy Secure Ciphertext-Policy and Attribute Hiding Access Control Scheme Based on Blockchain. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5784–5798. [CrossRef]
37. Rahnama, S.; Gupta, S.; Qadah, T.M.; Hellings, J.; Sadoghi, M. Scalable, resilient, and configurable permissioned blockchain fabric. *Proc. VLDB Endow.* **2020**, *13*, 2893–2896. [CrossRef]
38. Lee, J.W.; Park, S. A Study on Performance Improvement of Hyperledger Fabric Through Batched Chaincode Message. In Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), Daegu, Republic of Korea, 22–25 September 2020.
39. Sun, Y.; Yan, B.; Yao, Y.; Yu, J. DT-DPoS: A Delegated Proof of Stake Consensus Algorithm with Dynamic Trust. *Procedia Comput. Sci.* **2021**, *187*, 371–376. [CrossRef]
40. Kaur, S.; Chaturvedi, S.; Sharma, A.; Kar, J. A Research Survey on Applications of Consensus Protocols in Blockchain. *Secur. Commun. Netw.* **2021**, *2021*, 6693731. [CrossRef]
41. Chen, R.; Tu, I.P.; Chuang, K.E.; Lin, Q.X.; Liao, S.W.; Liao, W.J. Endex: Degree of Mining Power Decentralization for Proof-of-Work Based Blockchain Systems. *IEEE Netw. Mag. Comput. Commun.* **2020**, *34*, 266–271. [CrossRef]
42. Bala, K.; Kaur, P.D. A novel game theory based reliable proof-of-stake consensus mechanism for blockchain. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, 4525–4549. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.