mathematics

MDPI

Review

A Review of the Key Technology in a Blockchain Building Decentralized Trust Platform

Weichu Deng †, Teng Huang *,† and Haiyang Wang

Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China
* Correspondence: huangteng1220@gzhu.edu.cn
† These authors contributed equally to this work.

Abstract: Currently, the trust mechanisms of various Internet application platforms are still built under the orders of centralized authorities. This centralized trust mechanism generally suffers from problems such as excessive power of central nodes, single point of failure and data privacy leakage. Blockchain is a new type of distributed data architecture with non-tamperability, openness and transparency, and traceability, which can achieve secure and trustworthy sharing of data without the participation of third-party authorities. The decentralized trust mechanism built based on the blockchain provides a new research paradigm with broad development prospects to solve the problem of establishing reliable information sharing under the environmental conditions of incomplete reliability in finance, healthcare, energy, and data security. In response to the issues exposed by centralized trust mechanisms in recent years, based on the critical technology of blockchain, this paper surveys the relevant literature around the vital issue of building a decentralized and secure trust mechanism. First, the decentralized trust mechanism architecture is sorted out by comparing different decentralized platforms. The blockchain is divided into the data layer, network layer, consensus layer, contract layer and application layer, which correspond to the theory, implementation, operation, extension, and application of the decentralized trust mechanism of a blockchain, a district-centric platform. Secondly, the principles and technologies of blockchain are elaborated in detail, focusing on the underlying principles, consensus algorithms, and smart contracts. Finally, blockchain problems and development directions are summarized in light of relevant literature.

Keywords: blockchain; decentralization; smart contract; consensus mechanisms

MSC: 68M14

Citation: Deng, W.; Huang, T.; Wang, H. A Review of the Key Technology in a Blockchain Building Decentralized Trust Platform. *Mathematics* **2023**, *11*, 101. https://doi.org/10.3390/math11010101

Academic Editor: Jan Lansky

Received: 19 November 2022
Revised: 14 December 2022
Accepted: 21 December 2022
Published: 26 December 2022

1. Introduction

The traditional centralized trust mechanism is endorsed by authorities but can easily cause unpredictable losses as the authority's credit declines. A centralized trust model often has problems such as excessive power of the central node, single-point failure, data privacy leakage, etc. The global financial crisis in 2008 first exposed the shortcomings of the centralized trust mechanism in the monetary field. For this reason, scholars have sought a new alternative trust mechanism, and Bitcoin was born in this background. In *Bitcoin: A Peer-to-Peer Electronic Cash System* [1], Satoshi Nakamoto proposed Bitcoin, an electronic trading system that does not rely on credit, which is a decentralized electronic trading system based on the cryptographic encryption algorithm. Each node of the system stores the ledger of the transaction and ensures that the transaction data are not tampered with by using the encryption algorithm and proof-of-work. The use of Bitcoin can achieve peer-to-peer value transfer between the two parties without the help of third-party institutions. According to established rules, the Bitcoin system is not owned by any institution and is maintained by all participants.

The successful application of Bitcoin in the monetary domain has driven research toward decentralized technologies. The technology behind Bitcoin, blockchain, has emerged

as a solution to achieve decentralized trust in other domains. The industry widely acknowledges that Satoshi Nakamoto first proposed the blockchain concept in 2008. Its core idea originated from the concept of the signature information chain presented by BayerZhang et al. [2] in 1993. The signature information chain is an electronic ledger consisting of digitally signed documents that can be easily verified to see if they have been tampered with. Being similar to the signature information chain, blockchain can essentially be seen as a decentralized distributed ledger that multiple parties jointly maintain [3]. According to the degree of decentralization and ledger openness, blockchains can be classified into public chains, private chains, and consortium chains. As a decentralized trust mechanism, blockchain is not a new technology but a combination of P2P networks, distributed ledgers, asymmetric encryption, consensus algorithm, digital signatures, smart contracts, and other technologies.

With the development of more than ten years, blockchain technology has evolved from generation 1.0 to generation 3.0 [4], and blockchain has been applied to more fields from the early cryptocurrency. Blockchain 1.0: Bitcoin was born to achieve a decentralized electronic currency trading system. The cryptocurrency represented by Bitcoin is also the main application of blockchain 1.0. Both parties can create and manage transactions using the Bitcoin script in the Bitcoin system. However, Bitcoin script is a stack-based, non-Turing, complete scripting language, and thus only supports simple transaction management and cannot support complex transactions. Blockchain 2.0: To enable blockchain to handle more complex transactions, blockchain 2.0, represented by Ethereum [5], introduces smart contracts to apply blockchain to programmable finance [6]. Smart contracts are a kind of contract term established in digital form. Once deployed, smart contracts cannot be modified and are enforced when triggered, making their execution more transparent and efficient than traditional contracts. In the blockchain, the smart contracts abstract the terms of the contract into a program code, which is written into the block after being agreed upon by each node. While the introduction of smart contracts allows a blockchain to handle complex transactions, it is still limited to the financial field. Blockchain 3.0: To apply blockchain to more areas and let it become a decentralized solution for other areas, Hyperledger Fabric [7], and EOS [8], as a representative of blockchain 3.0 technology, were born. In this generation, blockchains are mainly represented by decentralized applications (Dapps), which apply blockchains to various industries [9] and provide decentralized solutions for urgent problems, such as education [10], the medical industry [11], energy [12], and the Internet of Things [13].

1.1. Related Work

Blockchain is regarded as a representative technology of the next-generation Internet [14]. It has been widely addressed by scholars in recent years due to its characteristics of no data tampering, transaction traceability, decentralized trust, and multi-party consensus. Xu et al. [15] conducted cluster analysis on the review literature retrieved from the Web of Science to determine the main research topics of the current blockchain. Yli-Huumo et al. [16] summarized and analyzed the current research status of blockchain technology from a technical point of view and then pointed out that the current blockchain research work lacks an evaluation of the effectiveness of the scheme. Remigijus et al. [17] have historical background and a comprehensive overview of blockchain technology and compared current popular blockchain platforms, especially consensus protocols. Zheng et al. [18] conducted a comprehensive investigation of blockchain technology from the perspective of applications, provided a blockchain classification, and discussed the current blockchain technology challenges and the latest progress in addressing them.

1.2. Goal and Contribution

Unlike related work, this paper revolves around how to build decentralized security trust mechanisms, taking a survey and comparison approach to more than one hundred blockchain papers in the last decade. Hot papers in the last decade are reviewed in an

extensive comparative survey and analysis. According to blockchain technology's hierarchy order, we analyze the problems and solutions faced by each layer, and based on this, the research progress of blockchain technology is introduced. This paper's main contributions are as follows:

- Through a comprehensive overview of decentralization technology and blockchain, we sort out the blockchain decentralization trust mechanisms by layer and analyze the problems and solutions faced by each layer.
- The current research status of the core technologies of blockchain, such as consensus mechanisms and smart contracts, is compared and analyzed. The advantages and disadvantages of current core technologies and future development directions are summarized.
- By investigating the current applications of blockchain, we reveal the technical and security problems of blockchain technology and provide guidelines for blockchain technology research.

1.3. Layout

This paper is organized as follows: Section 2 provides a comprehensive overview of decentralization technologies. Section 3 of this paper provides a comprehensive overview of decentralization technology and the development of blockchain technology in recent years. Section 4 deeply analyzes the basic model structure of blockchains and expounds on the principle of decentralized trust achieved by blockchain. In Section 5, from the application's perspective, the consensus mechanisms based on proof class and voting class are introduced, and the advantages and disadvantages of each consensus mechanism and the improved algorithm are analyzed. Section 6 discusses smart contracts from the operation architecture, contract language, and contract platform perspectives. Section 7 introduces the typical applications of blockchain in various fields. Section 8 analyzes the problems faced by the current blockchain technology based on cases. Finally, the research status and development trends of blockchain technology are summarized in Section 9.

2. Review

2.1. Decentralized Web

Decentralized technology using a blockchain (especially Bitcoin) is known to the world. Still, blockchain technology also emerged after the birth of several different decentralized platforms that are different from the blockchain. As examples, take decentralized video in PeerTube, a decentralized blogging system; and Mastodon, PrPl, a decentralized social network. Most of these platforms implement the functionality of a particular traditional web application and can therefore be collectively referred to as decentralized web (DW).

- Solid [19]: Solid is a decentralized data storage platform. In Solid, all users' data is independent of the applications that create and consume that data. Developers can use the Solid protocol to read, write, and access control users' pod content. The data on Solid belongs to the user, and the user can freely decide where the data are stored and who can access the data.
- Mastodon [20]: Mastodon is a decentralized online blogging system run by a decentralized federation of independently run open-source software servers. The primary goal of Mastodon is to give back control of content distribution channels to users and avoid inserting other irrelevant ads or posts in the information feed.
- PrPl [21]: PrPl is a decentralized online social network that can guarantee user data security. For the user, his data are safe; the ownership of the data will not be lost with the user's social activities; and it supports fine-grained data sharing. For developers, PrPl provides a Datalog-based development language called SociaLite for data access control. Using SociaLite to write a small amount of code can realize the sharing of private social data among friends.

- PeerTube: PeerTube is a free and open source decentralized video sharing platform that uses peer-to-peer technology to reduce the load on a single server when watching videos.
- Diaspora [22]: Diaspora is a decentralized online social network. It is a network of independent Diaspora servers managed by individual users who allow Diaspora users' profiles to be hosted on their servers. Therefore, the degree of decentralization is not high.

The comparison of decentralized web platforms is shown in Table 1. Decentralized technologies other than the blockchain are mainly concentrated in the traditional Internet (especially social networks), and a decentralized data storage and access control system has been established around the protection of user data. This also reflects the current urgent need for data security. Non-blockchain decentralized platforms are constructed similarly: (1) Allow anyone to establish an independent server (instance) so that people can register and use it locally; (2) local instances of different users are integrated through alliance collaboration.

Table 1. Decentralized web technology comparison.

Platform	Appearance	Area	Technology
Solid	2016	Social Network	Linked Data
Mastodon	2016	Blog System	ActivityPub
PrPl	2010	Social Networks	Datalog
PeerTube	2018	Video Sharing	P2P
Diaspora	2012	Social Networking	Rails

2.2. Blockchain

Although DW applications can replace traditional Internet applications to a certain extent, such as Facebook, YouTube, Twitter, etc., these decentralized platforms are limited to specific fields, and the technologies they adopt are also different and fragmented. For example, Solid is based on linked data, and Mastodon is based on ActivityPub, so they are not widely used like blockchain. Unlike the messy system between DW applications, the blockchain is more systematic and complete and has received extensive attention and development in recent years.

In terms of the basic theory of blockchain: Dang et al. [23] aimed at the blockchain's scalability problem, applying sharding technology (a commonly used database expansion technology) to the blockchain. By designing an efficient sharding protocol and relying on trusted execution hardware, the sharded blockchain achieves high throughput, which is comparable to Visa-level performance; graf et al. [24] proposed a distributed ledger security analysis framework, Fledger, and conducted a security analysis on the non-blockchain distributed ledger Corda for the first time; xu et al. [25] proposed vChain, a novel blockchain framework for data-verifiable queries. The framework alleviates storage and computation costs for users and employs verifiable questions to guarantee the integrity of results. Yu [26] proposed a non-permission blockchain protocol OHIE, which uses the bandwidth of the network as simply as possible to improve the throughput of the blockchain. Ruan et al. [27] proposed LedgerView, which introduced an access control view system for hyperledger fabric and realized the management of access rights to sensitive information. Han et al. [28] designed Shrec, a novel transaction relay protocol for high-throughput blockchain systems. This realization shows that, compared with other protocols, this protocol can reduce the bandwidth consumption by 60% while increasing the system throughput by 90%.

In terms of the practical application of blockchains: sestrem et al. [29] applied blockchain technology to a smart grid to protect the privacy of the smart grid's user data; Al-Mamun et al. [30] applied blockchain to high-performance computing (HPC), proposing the first practical blockchain system for attestation services on HPC. The system incurs orders of magnitude less overhead than existing solutions while keeping data trustworthy; Kaur et al. [31] proposed the delegated proof-of-accessibility (DPoAC) consensus

mechanism, which makes the blockchain suitable for resource-limited networks by using technologies such as secret sharing, randomly selected PoS, and the Interplanetary File System (IPFS). Lansky et al. [32] proposed an ECC-based lightweight authentication protocol BCmECC in the IoT scenario, which can resist temporary information attacks.

As shown in Table 2, the current versioni of blockchain has received extensive attention and research both in theory and application. In theory, recent research mainly focuses on performance improvements for blockchains. The performance problem of the blockchain is also one of the main reasons restricting the wide-scale application of the blockchain. In practical applications, the blockchain's decentralized trust mechanism can solve data privacy and security problems, identity authentication, and high data reliability.

Table 2. Blockchain technology progress.

Author	Year	Category	Outcome
Dang et al.	2019	performance improvements	increased throughput
Graf et al.	2021	Security Analysis	First security analysis of non-blockchain ledger
Al-Mamun et al.	2021	Practical Applications	Applying Blockchain to High Performance Computing
Sestrem et al.	2020	Practical Applications	Applying Blockchain to Smart Grid
Kaur et al.	2022	Performance Improvements&Applications	Applicable to Resource-Constrained Scenarios
Lansky et al.	2021	Performance Improvement&Applications	Resistance to Temporary Information Attacks
Xu et al.	2019	Performance Improvements	Data Verifiable Queries
Yu et al.	2020	performance improvements	increased throughput
Ruan et al.	2022	performance improvements	access control implemented
Han et al.	2020	performance improvements	increased throughput

3. Blockchain Architecture

Around how to realize decentralized trust, a blockchain can be divided into five layers from a technical point of view, which are the data layer, network layer, consensus layer, contract layer, and application layer, as shown in Figure 1. The data layer defines the blockchain's underlying data structure, storage structure, and ledger pattern as the theoretical basis and outlines a theoretical model of decentralized trust. The theoretical model of blockchain decentralized trust in the network layer is realized by utilizing the distributed P2P network. In the consensus layer, the consensus algorithm organizes and coordinates the behaviors of nodes in the decentralized system to drive the continuous operation of the blockchain. In the contract layer, smart contracts are introduced as the extension of the blockchain so that the blockchain can handle more complex transactions. At the application layer, providing blockchain APIs makes it easy for developers to build Dapps and offer decentralized solutions to problems from various industries.

3.1. Data Layer

Due to the lack of authoritative central node coordination and management, the decentralized system has problems such as easy data tampering, untraceable node behavior, and difficulty in rapidly authenticating transactions, leading to the data not being trusted. As the theoretical basis of blockchain, the data layer needs to solve the appealing problem to ensure that the data are credible to achieve decentralized trust. From the perspective of the logical structure of data, the blockchain is a chain composed of a connected block, and each block stores the transaction information. The blocks are connected by hash pointers and are chained in chronological order of their generation. According to the characteristics of the hash function, any slight modification to the block data will create a huge change in the hash value of the block, leading to the block not being chained. Based on that, security ensures that the data on the block is not tampered with, and thus the credibility of the data on the chain. The data layer constructs the decentralized trust model of blockchain from three aspects: data structure, storage structure, and ledger pattern.

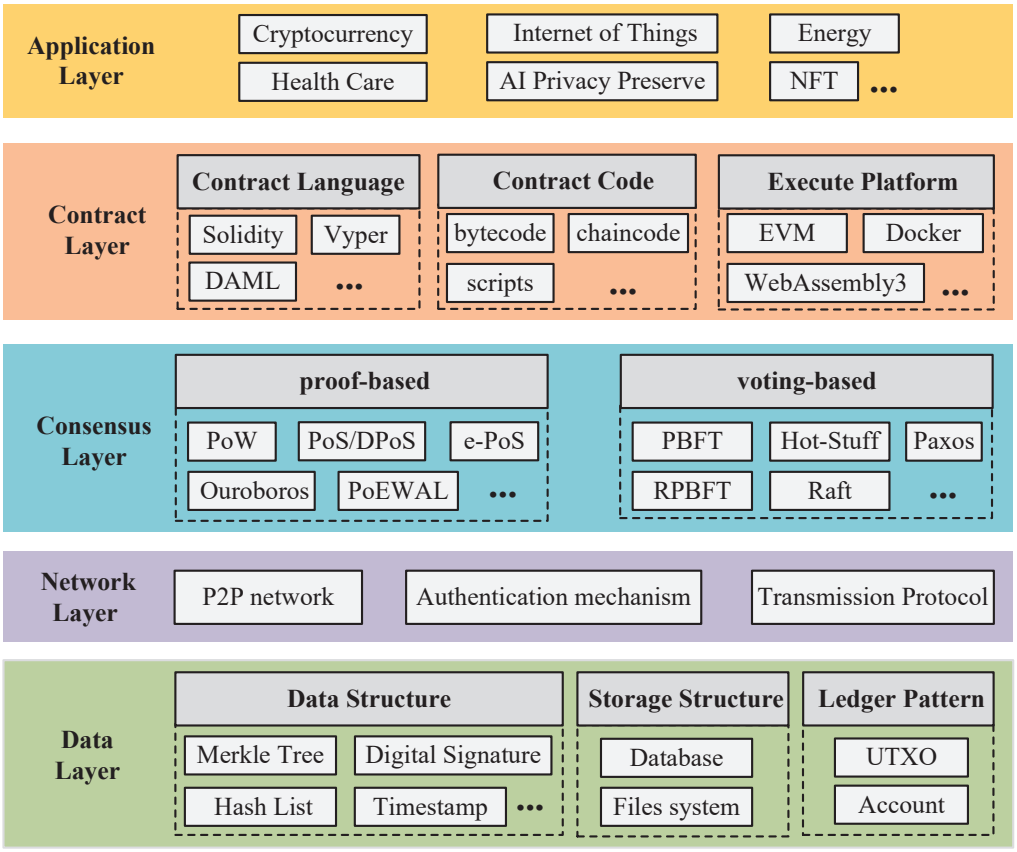

Figure 1. Blockchain architecture.

3.2. Network Layer

The network layer is the key to implementing a decentralized system at the physical level. Decentralization means that the blockchain nodes are peer-to-peer at the physical level and that each node can communicate with each other without passing through the central node. Therefore, the network structure of the blockchain adopts the decentralized P2P structure. As shown in Figure 2, compared to a centralized network structure, a P2P network can ensure peer-to-peer communication between nodes, and nodes can join or exit the system quickly.

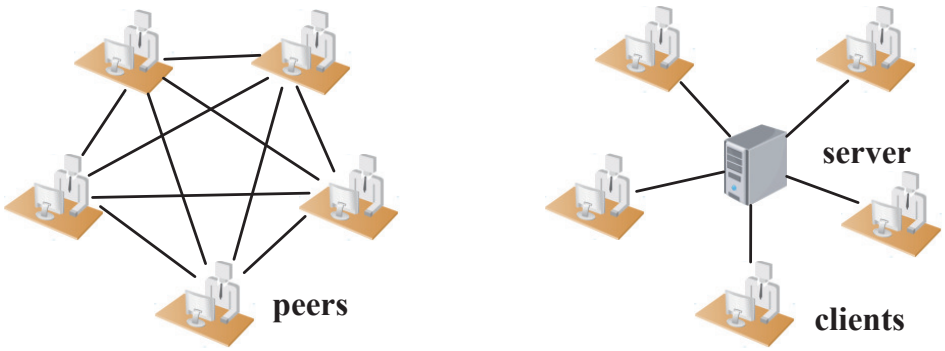

Figure 2. P2P network vs. centralized network.

The P2P network is a distributed application architecture. The P2P networks were initially designed to facilitate the distribution of large files over unreliable networks. In a P2P network, multiple computers are connected in a peer-to-peer position, and the entire network does not require centralized coordination by a central processing node. In P2P networks, each peer can act as both requestor and responder of network services. Research

on p2p network technology has recently focused on improving system performance and security. In system performance: Abudaqa et al. [33] summarized, evaluated, compared, and classified the techniques used to improve the performance of P2P file-sharing systems based on network coding; Milojicic et al. [34] provided a general analysis of the design and implementation issues of P2P systems in the context of practical cases. In security: Alharbi et al. [35] explored the security weaknesses and threats in P2P networks and proposed that the fundamental problem of P2P networks is the trusting of peers and the problem of secure traffic routing. Risson et al. [36] discussed the metrics affecting the robustness of P2P systems.

3.3. Consensus Layer

The consensus layer implements the consensus algorithm, which organizes and coordinates the decentralized system, allowing the blockchain to operate securely and stably. A blockchain is a distributed system where nodes communicate and coordinate with each other only through messaging because no central node is involved. In a distributed system, nodes agreeing on an event is also called a consensus, and a consensus algorithm is used to ensure data consistency among nodes in the system. Due to unavoidable problems such as network latency, node failure downtime, and bandwidth limitation, distributed systems are subject to the FLP impossibility principle and CAP theory. The FLP impossibility principle [37] means that in a system containing multiple deterministic processes, as long as one process may fail, no protocol can guarantee a finite time for all processes to agree. CAP theory points out that it is impossible for any distributed system to satisfy consistency, availability, and partitioning of fault tolerance at the same time [38], as shown in Figure 3. Therefore, according to CAP theory and the FLP impossibility principle, certain aspects must be traded off when designing consensus algorithms for blockchains.

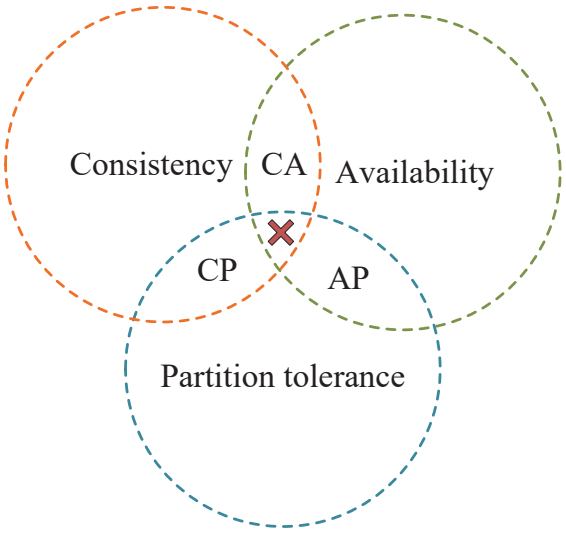

Figure 3. CAP theory.

Blockchain can be regarded as a distributed public ledger. The essence of consensus is to decide the bookkeeping right, i.e., to solve the problem of who can produce the blocks and package the transactions into the blocks. According to the different mechanisms to reach consensus, blockchain consensus algorithms can be divided into proof-based and voting-based.

3.4. Contract Layer

The contract layer implements smart contracts, a set of digitally set commitments that are unmodifiable once deployed and executed immediately once triggered. Smart contracts, as an extension of the blockchain, enable the blockchain to have the ability to handle logically complex transactions.

3.5. Application Layer

The application layer provides API interfaces for users to easily build Dapps using blockchain services and applies blockchains to various practical scenarios. With the development of blockchain technology, various Dapps have emerged to bring decentralized trust solutions to the problems of traditional industries.

4. Blockchain Basic Principle

4.1. Data Structure

A blockchain has a chain structure in terms of blocks to achieve data immutability. The data structure of different blockchain platforms differs in specific details but is the same overall. Take Bitcoin as an example. The block in Bitcoin is divided into the block header and the block body. The block header contains the version number, random number, hash of the previous block, Merkle tree root hash, timestamp, current workload proof difficulty, etc. The block body contains all the transactions packed into the block, and the Merkle tree comprises these transactions. To support smart contracts, Ethernet adds a system state to the block header for storing account balances, contract storage, contract code, and account random numbers.

A block contains a block header, timestamp, proof-of-workload random number, hash of the previous block, packed transactions, Merkle tree, etc. [1]. The block's verification signature and proof-of-work use cryptographic algorithms such as elliptic curve encryption and SHA-256. The data-layer structure differs slightly from blockchain platform to blockchain platform because of the different functions they focus on. Take the Bitcoin system as an example, and the data-layer structure is shown in Figure 4:

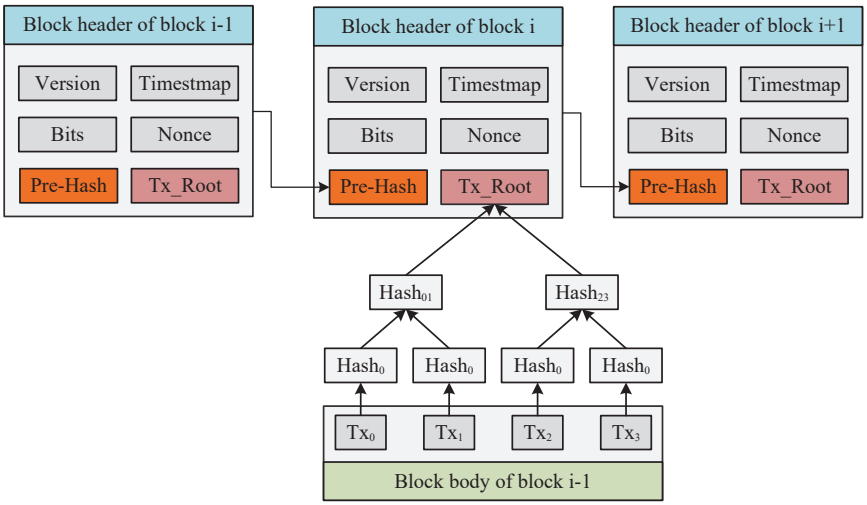

Figure 4. Blockchain data-layer structure.

To reduce the bandwidth consumption caused by block synchronization, each block in the Bitcoin system can be divided into two parts: the block header and the block body, which stores all the transaction records in the current block. Bitcoin nodes are divided into full nodes and light nodes. Bitcoin light nodes only need to synchronize the block header for block synchronization. The transaction records in the Bitcoin system are similar to the transaction records in the physical system. Each transaction record includes information such as the input and output addresses of the transaction information and the number of transfers. Based on this transaction information, a corresponding form of Merkle-tree structure can be generated from the bottom up. The hash value of the root node of the Merkle tree is stored in the header of the block, and at the time of each block generation, the bookkeeper of the block adds a timestamp to the block, which is used to mark the generation time of the block. As the timestamp is enhanced, the block is extended to form a chain of blocks with a time dimension, allowing data information to be traced back in

time. In addition, the block header contains the hash value of the previous block header, the version number, the random number of the proof of work, and the target hash value, among other information. Finally, the all information in the header of this block is hashed, and the resulting hash value exists in the header of the next block, which, in terms of logical structure, makes each block linked together in the form of a chain.

4.1.1. Hash Function

Hashing converts data of any length into a number within a fixed range. The conversion method is called a hash function, which calculates the value obtained after the original value is called a hash value. Take MD5, a widely used hash function, as an example. The MD5 algorithm is also called the MD5 message digest algorithm, which can generate a 128-bit hash value to ensure the integrity and consistency of information transmission. The MD5 algorithm is universal, stable, and fast; and it is widely used in the encryption and protection of ordinary data.

Hash functions are the basis of crucial blockchain technologies such as hash lists, digital signatures, and Merkle trees. The calculation of the hash function is unidirectional. It is easy to calculate the hash value of the given data, but it is difficult to deduce the original data given the hash value. The generated hash value may be the same for different data, and this phenomenon is called a hash collision. Due to the one-way nature of the hash function, people who want to generate hash collisions can only continuously try random numbers through brute force. Therefore, the process of finding suitable random numbers to create hash collisions is often used as "proof of work" by the blockchain.

4.1.2. Hash List

In order to ensure that the block data cannot be tampered with, the hash value of the previous block is retained in other blocks except the Genesis block, and the blocks are connected with the hash value to form a hash list. A hash list is a one-way chain table in which hash pointers connect nodes. Any small change in the block data will cause a huge change in the hash value, so it is impossible to tamper with the data in the hash list.

In addition to a chained structure, some scholars have proposed a blockchain with a non-chain structure for dealing with different scenarios. Qi et al. [39] proposed a cascade structure of blockchain to solve the performance problem of blockchains, which can accelerate the generation of blocks, expand the capacity of blocks, reduce the risk of bifurcation, and increase the security. Ribero et al. [40] proposed a cryptocurrency called DagCoin based on DAG structure, the first blockchain-based on DAG. DagCoin has no fixed blocks; each transaction has its own proof of work. The system can achieve a speed comparable to Bitcoin. Despite the emergence of blockchains with non-traditional chain structures, such as DAG and cascade structures, mainstream blockchains are still dominated by chain structures.

4.1.3. Timestamps

To make transactions traceable, Bitcoin adds timestamps to blocks and calculates the block's hash value by using the timestamp as the information in the block together. The timestamp is the total number of seconds from 00:00:00 GMT on 1 January 1970 to the present, and the timestamp proves that the transaction in the block must have existed at that time.

The current development of timestamps mainly revolves around improving timestamp accuracy and reducing errors. Zhang et al. [41] proposed an accurate blockchain-based timestamping scheme which solves the problem of the inaccuracy of file timestamps caused by blocks due to the existence of time errors in timestamps. Ma et al. [42] proposed an optimized blockchain timestamping mechanism that reduces the range of timestamps in blocks to an average of 10 min by serving external trust timestamps to the blockchain consensus.

4.1.4. Merkle Tree

Blockchain stores all the transaction records of history, and the data volume of historical transaction data will become larger and larger as time goes by. It is unrealistic to verify the existence of a certain transaction by traversing all the historical transactions. To enable fast transaction verification, all transactions in the block are stored as a Merkle tree.

A Merkle tree is a tree that connects parent and child nodes with a hash pointer. Bitcoin uses the simplest binomial Merkle tree to quickly verify whether a transaction exists in a block. The structure of a binary Merkle tree is shown in Figure 5. Each leaf node in the tree corresponds to a SHA256 hash of one transaction data within the block. The value of the parent node is obtained by concatenating the values of the two child nodes and then performing a hash operation. Hashing between nodes is performed repeatedly until the root hash value is reached, when the transaction Merkle root is generated. The Merkle root is used to detect any tampering with the transaction data in the block, so as to ensure the integrity of the transaction data in the block.

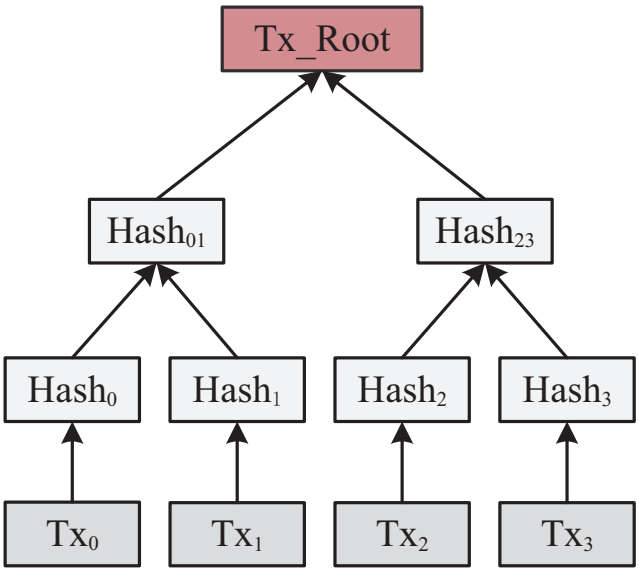

Figure 5. Merkle tree.

4.1.5. Digital Signature

Bitcoin is a chain of digital signatures designed to prevent transactions from being forged or denied. A digital signature is an unforgeable string of numbers that can be generated only by the sender of the message. It proves the validity of the sender of the message. Digital signatures are often used to verify the integrity of documents or messages and are an effective way to make transactions non-repudiation and unforgeable. In the process of Bitcoin transactions, the owner of a Bitcoin transfers the coin to the next owner by digitally signing it with the hash of the previous transaction and the next owner's public key and adding it to the end of the coin. The recipient can verify these signatures to validate the ownership of the coin.

Digital signatures are based on asymmetric encryption, first proposed by Rivest et al. [43]. Asymmetric encryption has two keys, which are used in the encryption and decryption processes. The commonly used asymmetric encryption algorithms in blockchain are RSA, SHA256, ECC, etc. As a decentralized distributed system, blockchain needs to adopt a compatible encryption algorithm because the system configuration of each node is different. RSA algorithm is an international standard algorithm that is widely used and compatible and can be applied to different systems. RSA is the first algorithm that can be used for encryption and digital signature, and it is also considered one of the best public key schemes. Although RSA has the characteristics of strong compatibility and high security, RSA has the problems of long key and time-consuming cryptographic computation. Compared to

RSA, ECC has the advantages of small key length, high-security performance, and small time consumption for the whole digital signature. Compared with RSA, ECC can use a shorter key to achieve comparable or higher security than RSA.

4.2. Storage Structure

During blockchain transaction execution, transaction data need to be packaged into blocks, and data writing is in high demand. In the process of blockchain transaction validation, it is necessary to quickly locate the block where the transaction is in and perform transaction validation. Based on the above functional requirements, blockchain often uses a combination of file systems and databases to store block data. The file system can facilitate the system to append data in the form of logs, and the database stores the index information of the file where the block is located, which can quickly find the location of the relevant transaction block and assist the system in query. Block data and block "undo" data are stored in the file system, and block "undo" data are the data for rolling back the blockchain when the system generates a chain fork. The database stores the state and index data of the blockchain, which are usually stored in key–value pairs for quick querying.

4.3. Ledger Pattern

A blockchain is a decentralized transaction ledger, and the ledger records the history of all transactions. There are two main types of mainstream ledger patterns: transaction-based and account-based.

4.3.1. Transaction-Based Ledger

The transaction-based ledger is used for digital currency transactions and is the ledger model used by Bitcoin. In Bitcoin, an "Unspent Transaction Output" (UTXO) is used instead of a centralized institution to clear transactions. In this transaction-based model, the user's assets are not explicitly recorded directly in the system but instead extrapolated from the information in UTXO. In order to know how many bitcoins a user has in total assets, we need to calculate how many coins that user has in total in all accounts in UTXO. The transaction-based ledger can record each transaction, trace the origin of each fund, and protect user privacy.

4.3.2. Account-Based Ledger

The account-based ledger is suitable for blockchain platforms that support smart contracts, such as Ethereum and Hyperledger Fabric. The account-based ledger model is similar to a bank account, where the account balance information is recorded explicitly by the system, and the transaction balance and business status data can be easily checked. Take Ethereum as an example. Ethereum accounts are divided into external accounts and contract accounts. External accounts are controlled by public–private key pairs; the user locally generates a public-private key pair. The private key controls the account also called a normal account. The user creates a contract that returns an address, and the contract can be invoked as long as the address of the contract is known. The account-based ledger gives participants a more stable identity and better support of smart contracts.

5. Consensus Mechanisms

In the decentralized scenario, without the participation of the central node, a fair operation mechanism, i.e., a consensus mechanism, must be established among the nodes of the blockchain to enable each node's unified and coordinated operation. Blockchain establishes a "trustworthy" network among nodes through the consensus mechanism so that each node can reach an agreement and achieve data consistency in the ledger of each node in the blockchain, which drives the continuous operation of the blockchain. The consensus mechanism of blockchain mainly solves the problem of who will construct the block and who will package the transactions into the block [44]. The consensus mechanism is the core of blockchain technology, which determines the security, scalability, and distributed nature

of blockchain system. The problem of consensus originates from the "biliteracy problem", and later the "Byzantine general problem" was proposed. The biliteracy problem refers to how to achieve reliable communication over unreliable channels. The Byzantine problem refers to the problem of how to make a distributed system agree in the presence of malicious behavior (e.g., message tampering or forgery), and the nodes that can both fail and behave badly are called "Byzantine nodes". Consensus algorithms can be divided into classical distributed system consensus algorithms and blockchain consensus algorithms, depending on the time. Classical distributed consensus algorithms include Paxos, Raft, and Kafka. According to the different mechanisms used to reach consensus, blockchain consensus algorithms can be divided into proof-based and voting-based. The proof-based consensus algorithms require "some competition" among nodes to decide the bookkeeping rights, such as proof-of-work (PoW) and proof-of-stake (PoS). Proof-based consensus algorithms do not require the strict identity of participants, and nodes are free to join and exit, so proof-based consensus algorithms are commonly used in public chains. The voting-based consensus algorithm is initiated by a node to reach consensus by having the whole network nodes vote on whether to agree to the proposal, such as practical Byzantine fault tolerance (PBFT) [45]. The voting-based consensus algorithms require a high identity of participating voting nodes and control the joining and exiting of nodes by the access mechanism, so voting-type consensus algorithms are commonly used in consortium chains.

The consensus algorithm is the core of blockchain technology and a research hotspot of blockchains. The current research on consensus algorithms mainly focuses on two aspects: performance optimization and application. In performance optimization: Wu et al. [46] proposed a hybrid consensus algorithm for blockchains that combines the advantages of PoS and PBFT algorithms. It reduces the number of consensus nodes to a fixed value through verifiable pseudo-random ordering and witnesses transactions between nodes. The improved hybrid consensus algorithm has excellent scalability, high throughput, and low latency, which is superior to the previous single algorithm. In applications: Biswas et al. [47] proposed a proof-of-block-transaction (PoBT) consensus algorithm. The algorithm allows the verification of transactions and the reduction of computation time for blocks, improving the performance of the system in terms of security, computation, memory, and bandwidth. Fu et al. [48] proposed a framework for evaluating consensus algorithms to provide guidance for the selection of consensus algorithms in sundry blockchain application scenarios.

5.1. Pow

5.1.1. Overview

The proof of work (PoW) algorithm is one of the most widely used consensus algorithms in blockchain systems, and the Bitcoin system uses the PoW consensus algorithm. The PoW algorithm was first used for spam detection [49], and the core idea is to include in the email proof that a certain job has been completed (hence the name "proof of work"). Usually, the calculation of such proofs takes a few seconds, so this does not cause any difficulties for casual users. However, for spammers, this can take weeks to send millions of spam emails. Email recipients can easily verify if an email is a spam by proof of workload.

In the blockchain using the PoW consensus algorithm, nodes need to constantly search for a specific random number, which is usually required to be calculated by a hash function (e.g., SHA-256) to obtain a hash value starting with several zero bits. It can be verified that the average work required to compute the random number is an exponent of the number of required zero bits. Due to the one-way computation and irreversible nature of the hash function, the random number found by the node is easily verified. In the Bitcoin system, the first node to find a specific random number is given bookkeeping rights and 50 coins as a reward. Hence, the process of finding random numbers is also called: "mining".

5.1.2. Advantages and Disadvantages

The Bitcoin system has been running smoothly since its launch in 2009 without any major failures, which is a testament to the effectiveness and security of PoW. In the PoW consensus, a node needs to control 51% of the computing power of the whole network to launch an attack. In the absence of a centralized node, the probability of a successful node attack is very low. Therefore, PoW consensus can effectively guarantee the security of the blockchain system. However, in PoW consensus, nodes constantly performing hashing operations will consume a large number of power resources, and blockchain chain systems using PoW consensus generally have serious energy consumption problems. In addition, the throughput of transactions in PoW consensus is very low. E.g., Bitcoin processes about seven transactions per second due to the limitation of block-out time and block size. This low transaction throughput makes it difficult to meet other application scenarios.

5.1.3. Improved Algorithms

PoEWAL (proof of elapsed work and luck) [50]: The PoEWAL consensus reduced the energy cost of the consensus by adding a time limit to the PoW. The mechanism emphasizes consensus by solving problems partially rather than completely within a fixed time frame. By adjusting the size of a given time period, the resource consumption of block mining can be effectively reduced, and devices with low computing power can also participate in mining. However, the essence of the consensus is still to obtain more consecutive zero hash values through continuous hashing operations. There is a problem similar to PoW where nodes with high arithmetic power have a higher probability of successful mining.

The trust-based PoW mechanism [51]: It can effectively solve the problem of high energy consumption in PoW consensus while ensuring the security of the blockchain network. By introducing the attribute of the node credit value, the higher the credit value, the lower the difficulty of node mining. Using a malicious behavior detection mechanism, the behavior of nodes is divided into positive and negative aspects, and positive behavior helps to increase the credit value of nodes. In contrast, negative behavior decreases the credit value of nodes. The positive aspect is expressed as the number of valid transactions calculated and verified by the node in the consensus process. In contrast, the negative aspect is determined by the node's malicious behavior time and penalty coefficient, where the malicious behavior is divided into two types. One is the node's lazy inaction in the consensus process. The other is the node's double spending attack in the transaction. The system dynamically adjusts the penalty factor according to the actual malicious behavior of the node, but it will bring on an additional computational overhead for malicious behavior monitoring.

5.2. Pos/Dpos

5.2.1. Overview

The Proof-of-Stake (PoS) [44] algorithm is designed to solve the problem of wasting a lot of resources by using PoW mining. Unlike PoW, which determines the bookkeeping right through the arithmetic power of nodes, PoS differentiates the bookkeeping right through the "equity" of nodes owning coins. The core idea of PoS is that in a decentralized network, the node with the largest equity will have a greater incentive to maintain the network. In terms of implementation, PoS introduces the coin age to dynamically adjust the mining level of nodes with different equity. The older the node, the lower the difficulty of mining it. Based on the appealing advantages of PoS, the PoS algorithm was first adopted in the blockchain platform peercoin, and Ethereum's consensus mechanism was transitioned from PoW to PoS [52] on 15 September 2022.

Although PoS solves the energy consumption problem of PoW, the performance is still not improved. In response to the performance problems of PoS, Dan [53] proposed the delegated proof-of-stake (DPoS) algorithm [54]. The topology of DPoS is shown in Figure 6. DPoS reduces the pressure on the network by reducing the number of participating consensus nodes and adding an election mechanism to PoS. As a variant of PoS, DPoS is

similar to PoS in that the number of representative members is limited and elected by all, and the elected representatives participate in the consensus.

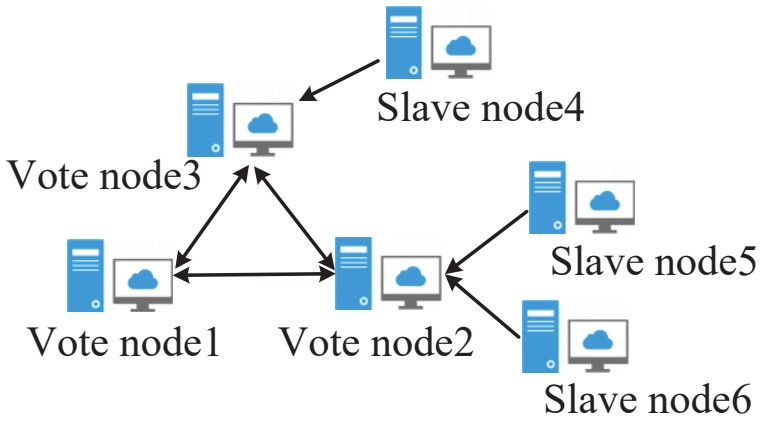

Figure 6. DPoS topology.

5.2.2. Advantages and Disadvantages

PoS consensus can significantly provide the transaction throughput of the system and reduce the energy loss in the consensus process. However, PoS consensus has disadvantages such as poor fairness and ease of generating the Matthew effect. The use of coin age will make it easier for the node with more tokens to gain bookkeeping rights, shifting the power gradually to that node, decentralizing the degree of decentralization, and making fairness worse.

5.2.3. Improved Algorithms

e-PoS [55]: In response to the possibility that PoS can lead to centralization and unfairness in blockchain systems, Saadd et al. improved PoS and proposed modular e-PoS. Compared with PoS, e-PoS can resist the power concentration of the network.

Ouroboros [56]: Kiayias et al. proposed the first proof-of-stake-based consensus protocol with strict security guarantees. Ouroboros also employed a new incentive mechanism to incentivize "proof-of-stake" protocols, where honest behavior is an approximate Nash equilibrium.

5.3. Pbft

5.3.1. Overview

Practical Byzantine fault tolerance (PBFT) [45] can tolerate Byzantine faults. The PBFT algorithm was proposed by Miguel Castro and Barbara Liskov in 1999. It improves the efficiency of the Byzantine algorithm and reduces the complexity from exponential to polynomial, making Byzantine fault tolerance practical. The PBFT algorithm can achieve $2f + 1$ fault tolerance; f is the number of Byzantine nodes that can be tolerated; and $2f + 1$ can ensure that the correct nodes in it send more information than malicious nodes. Therefore, the minimum number of nodes required by PBFT is $3f + 1$ (the maximum number of fault-tolerant nodes is $(n - 1)/3$).

The PBFT algorithm is divided into five stages: request, preparation, preparation, confirmation, and reply. The process is shown in Figure 7. In the request stage, the client initiates a transaction request to the master node. In the pre-preparation phase, the master node verifies the message signature after receiving the request from the client. After the message signature verification is passed, it broadcasts the pre-preparing message to all the network's nodes. In the preparation phase, the replica node verifies the message after receiving the pre-preparing information broadcast by the master node. If the verification is passed, the node broadcasts the prepare message to other nodes. In the confirmation phase, after receiving the correct prepare message from 2f other nodes, the node will enter the prepared state and send a commit message to other nodes. In the reply phase, after

the node receives the commit message, it verifies the message, passes the verification, and waits for the commit message sent by 2f + 1 different nodes. After receiving the message, it will send a reply message to the client.

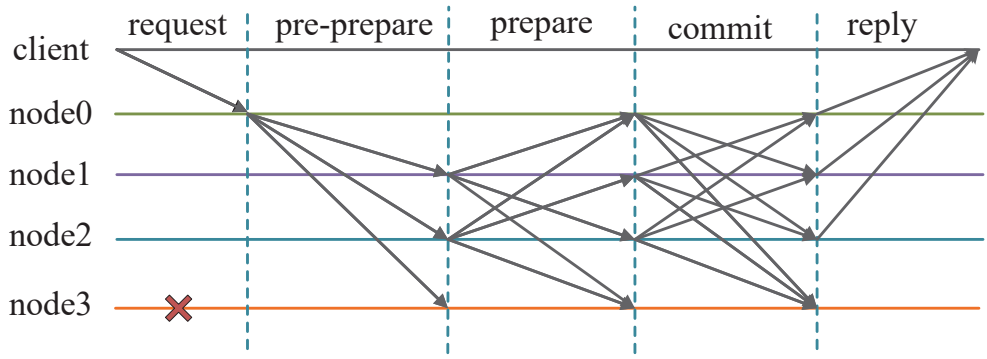

Figure 7. PBFT consensus process.

5.3.2. Advantages and Disadvantages

The PBFT algorithm can realize Byzantine fault tolerance with polynomial complexity and reach a consensus in the presence of malicious nodes in the network so that the Byzantine fault tolerance algorithm can be applied in practical systems. However, the PBFT algorithm has problems such as high communication complexity, a fixed number of nodes, poor scalability and dynamics, and only being suitable for private chains or consortium chains. In terms of network resource consumption, the frequent broadcasting of messages by the system will also lead to high bandwidth consumption. When the number of participating nodes increases, network congestion will likely occur, resulting in system performance degradation. Regarding the number of participating nodes, the number of nodes in the PBFT algorithm remains unchanged, the nodes cannot enter and exit at will, and the number of nodes is fixed.

5.3.3. Improved Algorithms

Hot-Stuff [57]: This algorithm was proposed by Abra et al. It improves the efficiency of the distributed consistency algorithm by making improvements to PBFT. The Hot-Stuff algorithm uses a parallel pipeline processing proposal, which is equivalent to combining the preparation and commitment phases in PBFT into one phase. In addition, Hot-Stuff uses linear view change (LVC), which reduces the communication complexity in view change.

RPBFT [58]: In response to the problems of arbitrary master node selection, a high communication overhead, poor dynamics, and low efficiency in the PBFT algorithm, Li proposed the practical Byzantine fault-tolerant consensus algorithm (RPBFT) based on role management. The RPBFT algorithm divides nodes into three roles, manager, candidate, and normal nodes; and realizes the transition between roles through a reward mechanism and election mechanism. Each role has specific responsibilities, so the nodes do not need to restart the system during joining and exiting. Meanwhile, using a synchronous verification mechanism instead of the traditional view replacement protocol increases the node efficiency.

5.4. Discussion

The consensus mechanism is the core part of a blockchain. The traditional distributed consensus mechanism (PBFT) is not well adapted to the unique open environment of the blockchain, and the network connection is replicated between nodes. Therefore, traditional distributed consensus blockchain systems often employ various networking assumptions. However, reality often differs from our assumptions. Consensus mechanisms explicitly designed for blockchains (such as PoW, although its original purpose is not this, are still regarded as representatives of blockchain consensus mechanisms) often do not need to make various assumptions about the network and nodes. Thus, openness and decentralization

tend to be stronger. In different application scenarios, the two have their advantages and disadvantages, and blockchain designers must choose.

6. Smart Contracts

Smart contracts are the core of blockchain 2.0 [59], represented by Ethereum smart contracts. They allow a blockchain to handle complex transactions not just limited to cryptocurrency ones. The concept of smart contracts was proposed before the emergence of blockchain, almost simultaneously with the emergence of the modern Internet. However, limited by the technological development at that time, smart contracts were not widely used until the emergence of blockchains.

Smart contracts are digitally established contractual terms that are self-verifying, self-executing, and do not require a third party. Compared to traditional contracts, smart contracts are more efficient, less costly, more secure, and free from "repudiation". Smart contracts are designed to perform safely and efficiently without a trusted third party, which aligns with the "decentralized trust" of blockchain. The smart contract in a blockchain is essentially a piece of code that runs continuously, cannot be modified once deployed, and is executed automatically when a predefined condition is triggered. A blockchain enables reliable information exchange, value transfer, and asset management through smart contracts.

6.1. Development

Smart contracts were first proposed by American computer scientist Nick Szabo in 1995 [60]. In 2009, the Bitcoin platform went online, supporting the use of Bitcoin scripts to manage transactions with the prototype of smart contracts. Bitcoin also represented the first generation of blockchain technology. In 2014, Ethereum [61] introduced smart contracts and supported the creation of smart contracts in the Turing-complete programming language. In 2016, Kosba et al. [62] proposed Hawk, a smart contract development framework that protects user privacy. In 2018, Kalra et al. [63] proposed ZEUS, a smart contract security analysis framework. The framework provides an order of magnitude improvement in security analysis time compared to previous techniques. In 2020, Zheng et al. [64] classified smart contract applications by comparing and analyzing typical smart contract platforms.

In summary, smart contracts are evolving towards easier development, higher security, and widespread application. Additionally, with the rise in blockchain technology, smart contracts will also receive more attention from scholars while developing rapidly.

6.2. Contract Languages

Smart contracts are deployed to blockchains, which requires the contracts to be strongly typed, as blockchains have valuable storage space. In addition, smart contracts should be easy to read and not misleading. Therefore, traditional programming languages such as C/C++ and Java do not write smart contracts very well. Programming languages for smart contracts have been born to meet the development needs of smart contracts.

6.2.1. Solidity

Solidity [65] is a new language developed specifically for Ethereum smart contracts. It has a syntax similar to JavaScript and runs on EVM. Solidity is a statically typed programming language that supports inheritance, libraries, and user-defined types. It can be used to create voting, crowdfunding, blind auctions, and multi-signatures. It can be used to create a variety of contracts, such as voting, crowdfunding, blind auction, and multi-signature wallet. On Ethernet, solidity contracts are compiled into bytecode, written to blocks through special transactions, and eventually executed by other transactions driven by the Ethernet VM. Solidity is one of the most widely used contract languages today, but at the same time, solidity has seen many security vulnerabilities and corresponding attacks.

6.2.2. Vyper

To solve solidity's security vulnerabilities, Vyper provides a smart contract language focusing on simplicity, suitability, and security [66], a contract-oriented Python programming language targeting EVM [67]. Vyper has a very clean and easy-to-understand syntax, so it is almost impossible for developers to write misleading programs.

6.2.3. Daml

The DAML language [68] is a domain-specific language specifically designed to encode shared business logic for simple, secure, and efficient applications. DAML is used for developing and deploying distributed applications in blockchain environments and is one of the best programming languages for smart contracts. Developers can use DAML to write applications quickly and concisely as an open-source programming language.

6.3. Platform Comparison

6.3.1. Bitcoin

In the Bitcoin network, users can write Bitcoin scripts to manage transactions. Bitcoin scripts are used to implement bitcoin transaction validation by checking a transaction's lock script and unlock script. Bitcoin scripts are stack-based, non-stateful, non-Turing-complete scripting languages with no complex statements such as select and loop statements, and therefore, they have limited functionality. Bitcoin scripting reduces the complexity of the system while meeting the requirements of transaction needs. However, it also brings disadvantages, such as low flexibility and limited usage.

To allow Bitcoin to adapt to different systems, Bitcoin scripts are designed to be stateless so that a script can be executed similarly on any system. Suppose a script is validated on one system. In that case, it ensures that every other system in the Bitcoin network can also validate the script, meaning that a valid transaction is valid for everyone. A Bitcoin script is a sequence of actions for a transaction that describes what happens to the next person who wants to spend the bitcoins being transferred and will gain access to them, divided into locking scripts and transaction scripts. Bitcoin scripts have the makings of a smart contract.

6.3.2. Ethereum

For the first time in a blockchain system, Ethereum introduced smart contracts that support Turing completeness [5]. Ethereum uses Solidity to write smart contracts. Solidity is a contract-oriented, high-level programming language created to implement smart contracts. In Ethereum, smart contracts deploy bytecode to the Ethereum network through transactions. Ethereum successful deployment generates a new smart contract account, executed by an Ethereum Virtual Machine (EVM). When deploying a smart contract, the contract code is first compiled into EVM bytecode by the SOLC smart contract compiler, and then a single transaction is used to create the smart contract. Ethereum smart contracts are Turing-complete, so in theory, users can write programs that do anything with them. It is easy to create contracts for voting, crowdfunding, closed auctions, multi-signature wallets, etc., using solidity, and they can meet most smart contract development needs.

6.3.3. Hyperledger Fabric

Hyperledger Fabric [7] is a platform for distributed ledger solutions based on a modular architecture that is highly confidential, resilient, flexible, and scalable. Its main purpose is to support the pluggability of different components to the complexity and complexity of the economic ecosystem. Hyperledger Fabric typically deploys smart contracts in the form of chain code. In Hyperledger Fabric, the chain code is the business bearer and is primarily responsible for the specific business logic, i.e., encapsulating transaction definitions and processing logic into interfaces. Each chain code runs in a protected container (Docker), isolated from the running of background nodes. Hyperledger Fabric supports writing

smart contracts in multiple languages, such as golang, java, and node.js, which greatly reduces the development threshold for smart contracts.

6.3.4. Eos

The Enterprise Operation System (EOS), a commercially distributed application blockchain operating system, is a new blockchain system developed by Block.one which aims to decentralize everything. As a new blockchain architecture [8], EOS provides a platform for smart contract development. It distributes storage designed to address scalability issues common in blockchain systems such as Ethereum and Bitcoin. EOS provides a decentralized application development environment with high transaction throughput through dPoS consensus and BFT consensus. Unlike Ether, which uses a virtual machine to execute smart contracts, EOS uses WebAssembly3, a portable, small, fast-loading, and web-compatible format, so users can write smart contracts in various languages as long as they can be compiled into WebAssembly3 (e.g., C++).

6.3.5. Avalanche

Avalanche is a new generation of public chain projects, and the main network was launched in September 2020. Avalanche is not a blockchain but a collection of blockchains composed of multiple subnets. The subnet has a special subnet consisting of three blockchains, the Primary Network. The three chains are the exchange chain (X-chain), platform chain (P-chain), and contract chain (C-chain). Each of the three chains has its functions, and they can be converted across chains, making it more convenient for users to take advantage of assets. The X-chain is responsible for the establishment and transferal of assets, and most users use this chain when transferring assets or trading assets. The P-chain is responsible for storing the data, information, and verification work on the chain. The C-chain is responsible for the functions of smart contracts. This chain is compatible with EVM, so it can be applied to most smart contracts. Thanks to its unique structure compared with traditional blockchain platforms, Avalanche has higher performance—it can achieve more than 4500tps—and is more scalable and secure.

6.4. Example

The following is an example of a money transfer contract to show the complete workflow of smart contract development, deployment, and execution. Suppose A wants to transfer money to B through a smart contract. The contract workflow is shown in Figure 8. First is development, where the business process of transferring money from A to B is written as smart-contract source code, and the source code is compiled into bytecode by a compiler. Next is deployment, where the compiled bytecode is deployed to the blockchain network via a single transaction. After consensus in the P2P network, the contract address is returned for contract invocation. Finally, when the deployed smart contract triggers an execution condition or is invoked to execute the contract transaction (e.g., deducting a specified amount from A's account and adding a specified amount to B's wallet), the result of the execution will be written to the block.

In the process of transferring funds from A to B, the whole process is open and transparent without the intervention of a third party, and the results of the transaction execution are written to the blockchain and cannot be tampered with.

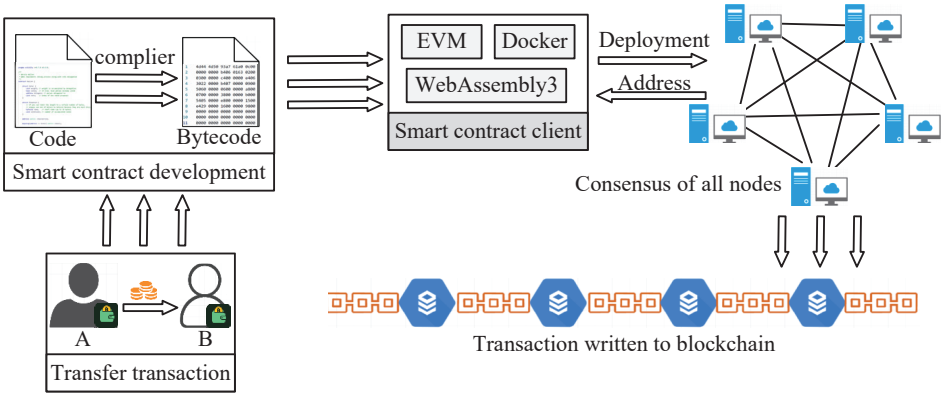

Figure 8. Smart contract workflow.

6.5. Discussion

The execution of smart contracts does not require the participation of a third party and can respond to user requests at any time, ensuring the fairness and efficiency of transactions. Before the contract is deployed, all the terms and execution processes have been formulated and executed under the computer's absolute control, so there is no possibility of errors in the entire process. Once the contract is deployed, all content cannot be modified. If one party breaks the contract, it will be punished accordingly. Using smart contracts can save transaction fees charged by banks and service fees of intermediaries. In addition to the advantages mentioned above, smart contracts still have the following problems: security issues, as it is difficult for anyone to guarantee the complete correctness of the code, and errors cannot be modified; interface problems, as each blockchain has different forms of storage for digital assets; the issue of how to call smart contracts across blockchains to realize asset transfers remains to be researched.

7. Applications

From blockchain 1.0 to blockchain 3.0, blockchain technology has been flourishing. Blockchain technology has also been applied from the earliest cryptocurrency to a wider range of fields, such as cryptocurrency, healthcare, IoT, Security AI, and NFT. [13]. The decentralized, open, and transparent characteristics of blockchain can also bring decentralized solution ideas to existing problems in some fields.

7.1. Cryptocurrency

Cryptocurrencies have been around since the 1990s but were not used and developed for various reasons until the emergence of Bitcoin made them widely known. Electronic cash (Ecash) emerged in 1990, changing the way traditional money works and allowing it to be traded digitally and anonymously over the Internet. In 1997, Back proposed the hashcash algorithm mechanism [69], which calculates a token through the CPU cost function and can be used as a proof of workload. In 1998, Dai proposed the electronic cryptocurrency system B-money, a distributed system that uses cryptography to control the currency for transactions, and first adopted the idea of decentralization to design cryptocurrency. In 2008, influenced by the global financial crisis, the international community began exploring innovative finance. Satoshi Nakamoto proposed Bitcoin in this context, which also marked the birth of Blockchain 1.0 technology. Satoshi Nakamoto combined a distributed system using cryptography from Ecash and B-money and a proof-of-work mechanism from Back and Finney to solve the trust and Byzantine problems. Bitcoin is a P2P form of digital currency. Unlike traditional currencies, Bitcoin does not have a central currency issuer, and the P2P network nodes work together to keep the system running. Bitcoin is also the most successfully used cryptocurrency to date.

Cryptocurrency is by far the most successful and well-known application of blockchain. Cryptocurrencies, represented by Bitcoin, were once synonymous with blockchain. It is

foreseeable that even in the future when blockchains are widely used, cryptocurrencies will remain among of the most important blockchain applications.

7.2. Energy

Current energy trading methods are still dominated by traditional centralized trading, which suffers from inefficient trading, opaque trading information, and long settlement times; and distrustful and opaque energy markets have potential security and privacy issues. In addition, intermittent energy sources and microgrids are an important part of the energy supply, and the increasing amount of renewables in the energy system requires new market approaches to pricing and decentralized generation [12].

Compared to centralized generation and single -arket pricing strategies, using a decentralized blockchain to control generation and energy trading can better incentivize generation organizations, improve generation efficiency, and facilitate energy trading. Kang et al. [70] proposed a localized P2P power trading system (PETCON) for local power trading among plug-in hybrid electric vehicles (PHEVs) based on consortium chain technology. In PETCON, electricity trading among PHEVs is resolved through an iterative double-auction mechanism that maximizes social welfare while protecting PHEVs' privacy. Su et al. [71] proposed a smart-contract-based energy blockchain system that enables secure charging services for electric vehicles by executing smart contracts. The experimental results show that the scheme has higher efficiency compared to other conventional schemes.

Blockchain technology will be applied more to decentralized energy management and energy trading in the future, and decentralized energy management systems can supplement the current centralized energy management system.

7.3. Healthcare

The current information systems of most medical institutions are centralized and stored independently, which makes it difficult to efficiently interconnect data among medical institutions and inconvenient for patients to seek medical treatment across institutions. Centralized information systems are also vulnerable to hacking and data leakage, compromising patients' privacy.

Blockchain's tamper-proof and verification features can ensure that patients' private information is not leaked [72]. Azaria et al. [73] have built a decentralized record management system (MedRec) to handle electronic medical data using blockchain technology. The system provides a comprehensive, immutable patient log and is easily accessible. Using PoW incentives enables patients to participate as "miners" in maintaining the system's security while allowing patients and providers to choose the release of metadata to facilitate medical research. healthbank, a Swiss global digital health startup, offers users a secure blockchain-based data management platform [74], where users can store and manage their health information data, and the sovereignty of the data is in the hands of the user. In addition, healthbank can act as a data trading platform where users can save data for medical research, and where users can receive specific financial compensation for the data they provide. hirtan et al. [75] implemented a medical data-sharing system using Hyperledger Fabric, which can share important information about medical analytics among hospitals, medical clinics, and research institutions based on patient-defined access policies. The system uses a combination of public and private chains to protect user privacy. The private chain stores the user's accurate ID information, and the public chain stores patient health information labeled with temporary IDs.

In summary, the use of blockchain to build a decentralized medical data management platform enables the sharing of medical data to facilitate medical research while ensuring the privacy and security of the data.

7.4. Internet of Things

IoT devices are found in various scenarios, such as cities, buildings, and homes. IoT combines various information sensing devices with networks to form a huge network to

achieve interconnection of people, machines, and things at any time and place, allowing traditional devices to become intelligent and autonomous [13]. However, the IoT still has issues such as security and privacy that hinder its widespread use.

A blockchain can establish decentralized trust in a distributed environment [76], which helps to overcome the security issues and privacy problems of IoT. Alphand et al. [77] combined an object-based IoT security architecture and an ACE authorization framework. Their solution uses a blockchain to replace a single ACE authorization server. It enables smart contracts, handles authorization requests, and uses a self-healing key distribution scheme to achieve efficient management of the IoT. Li et al. [78] proposed a multilayer, secure IoT network model based on blockchain technology, providing a wide-area network solution for the IoT. The model reduces the difficulty of blockchain deployment by dividing the IoT into a multi-layered decentralized network while ensuring the high security and trustworthiness of the blockchain. Pinno et al. [79] proposed a blockchain-based IoT access authorization architecture that ensures the privacy and confidentiality of information collected by IoT devices. The architecture is compatible with many access control models used in the IoT today.

In summary, more and more blockchain technologies are being applied to the Internet of Things (IoT) to solve the privacy and security problems in the IoT. However, a blockchain consumes many resources, and IoT devices generally have little computing power and storage space, so the traditional blockchain is not directly applicable to the IoT.

7.5. Security AI

Thanks to the development of computing power brought about by cloud computing and the generation of many samples in the era of big data, artificial intelligence technology, represented by machine learning, has been developed and used increasingly. However, studies have shown that [80] machine learning models are vulnerable to attacks that lead to privacy leaks, posing privacy and security risks.

Blockchain's data are highly redundant and decentralized, which is ideal for storing and protecting important privacy data from data loss or privacy leakage caused by attacks or mismanagement of centralized institutions. In recent years, various scholars have researched how blockchain can be applied to AI privacy protection. Zyskind et al. [81] implemented a decentralized personal data management system based on blockchain technology to ensure that users own and control their data. Additionally, they implemented a protocol that turns the blockchain into an automated access control manager that does not require a third party. Chen et al. [82] proposed LearningChain, a decentralized machine learning system for privacy protection and security, and designed a distributed stochastic gradient descent (SGD) algorithm to learn general prediction models. Decentralized SGD uses a differential privacy-based scheme to protect the data privacy of each party. Qi et al. [83] proposed a federated learning framework based on the consortium chains which can achieve secure and reliable federated learning without the need for a central model server. The federated learning framework can effectively protect model data privacy and prevent data poisoning attacks due to the noise-added differential privacy mechanism.

The blockchain can be regarded as a decentralized trusted database, replacing the centralized server to realize the data storage function required for machine learning and avoid privacy and security attacks on the central server.

7.6. Nft

A Non-Fungible Token (NFT) [84] is a token issued according to the Ethereum ERC721 and ERC1155 standards. It has indivisible, irreplaceable, and unique characteristics. Through NFTs, all tokenized properties can be freely traded with customized values based on age, rarity, liquidity, etc. NFT is mainly used for games, artworks, domain names, collectibles, virtual assets, real assets tokenization, and other fields, especially artwork and games that have received great attention in the market. NFT has greatly stimulated the prosperity of the decentralized application market. According to data from the cryp-

toslam website, as of August 2022, the cumulative transaction volume of NFT has reached $39,245,668,068. Wang et al. [85] conducted systematic research on NFTs for the first time, pointing out that the development of the NFT ecosystem is at an early stage, and related technologies need to be further developed.

7.7. Web 3.0

Web 3.0 is generally considered the next generation of the Internet, a decentralized Internet [86] running on blockchain technology. In this environment, users do not have to create multiple identities on different centralized platforms but can create a decentralized universal digital identity system that can pass through various platforms. The most prominent feature of Web 3.0 is that it can not only realize the exchange of data but also realize the circulation of value [87]. Web 1.0 data are read-only, such as Yahoo and MSN data. Web 2.0 data are read–write interactive, such as Facebook and Twitter data. Web 3.0 data are read–write interactive and owned and controlled by the creator; representative applications include Bitcoin, Ethereum, IPFS, etc. Web 3.0 is a new network infrastructure that integrates the traditional Internet, blockchain, programmable economy, etc. It is currently experiencing a blockchain, and its final architecture is uncertain, but the booming trend is unavoidable.

8. Blockchain Problems

8.1. Technology

8.1.1. Privacy

In a blockchain, in order to ensure that each distributed node quickly reaches a consistent consensus and thus achieves decentralized trust, all transaction records on the chain must be made public, and any node can access the data at any time, which in turn significantly increases the risk of user privacy leakage. Research shows that [88] malicious nodes can obtain users' privacy information by observing and analyzing the transaction records on the chain. Meiklejohn et al. [89] established the associations between bitcoin addresses and users' information by constructing a heuristic clustering model to cluster bitcoin wallet addresses and then classify users of these clusters using a re-identification attack.

8.1.2. Scalability

With the widespread application of blockchain technology, various chains have started to appear, but most of them are independent. It is difficult to have assets or data in different blockchain systems interact, and value silos are gradually formed by blockchains, which also underlines the problem of insufficient blockchain scalability.

8.1.3. Performance

In the actual blockchain applications, the performance drawbacks of blockchain systems are gradually being highlighted. Bitcoin, Ethereum, and other public chain projects are highly decentralized and commonly use proof-based consensus mechanisms represented by PoW. These consensus protocols are energy-intensive and inefficient. For example, Bitcoin can only process about seven transactions per second [90]. The user creating a transaction must wait an average of 10 min to ensure that the transaction is written to the blockchain, whereas Ethereum supports roughly 10–20 transactions per second. Consortium chains can avoid node mischief to a certain extent using access control and identity authentication of nodes in the network and adopt consensus protocols such as Raft and Kafka that only consider node collapse or network failure, giving specific performance improvements. Hyperledger Fabric [91], a representative project of the consortium chains, can process up to 2000 transactions per second. However, in comparison, mainstream companies of payment processing, such as Alipay and Visa, can process tens of thousands of transactions per second. The transactions can be confirmed within seconds, which shows that there is still a huge gap in the transaction processing capacities of blockchain

systems compared to traditional business systems. Despite the emergence of the lightning network [92], sidechain [93], and other technologies to enhance blockchain performance, blockchain performance still limits its large-scale application.

8.1.4. Lightweight

In order to build a decentralized trust mechanism, a blockchain needs to consume a large amount of computing and storage resources, and there are many limitations in applying traditional blockchains in resource-constrained scenarios. Regarding computation, nodes compete for bookkeeping rights through consensus mechanisms (e.g., PoW), a process that wastes a lot of computational resources and greatly consumes power resources. In terms of storage, nodes need to keep a complete copy of the blockchain to verify blocks and transactions. Due to the immutability of data, the blockchain ledger grows and accumulates over time. New nodes joining the network need to download huge amounts of ledger data, which greatly limits the participation of nodes with limited storage capacity in maintaining the blockchain and reduces the decentralization of the blockchain system.

8.2. Security

8.2.1. 51% Attack

The 51% attack means that if an attacker has more than 50% of the network's computing power, he can modify his transaction records, discard the blocks mined by the rest of the miners, and prevent transaction confirmations, among other malicious actions. In the Bitcoin network, the PoW consensus mechanism is used to solve the problem of obtaining bookkeeping rights, and the "longest chain consensus" is used to solve the problem of how to bookkeep. A 51% attack would be to take advantage of the Bitcoin network using PoW competition for bookkeeping rights and "longest chain consensus". The 51% attack takes advantage of the fact that the Bitcoin network uses PoW competition and "longest chain consensus" to generate a longer chain to "roll back" transactions that have already occurred. Therefore, in theory, if a node has more than 50% of the network's arithmetic power, it has the absolute advantage of gaining bookkeeping rights, can generate blocks faster, and has the right to tamper with the blockchain data.

8.2.2. Sybil Attack

Sybil attacks [94] are attackers manipulating or imitating multiple virtual identities on the blockchain. Sybil attacks often occur in P2P, wired, and wireless network environments. Sybil attacks formally generate as many identities as possible on behalf of the attacker's peers and behave like multiple peers in the system designed to interfere with the expected behavior of the system. The blockchain achieves the security and immutability of the network through redundant data in multiple nodes, and the witch attack weakens the role of redundant backup by controlling most of the nodes in the system. For example, in the anonymous voting of a blockchain project, the attacker attains much voting power by manipulating a large number of virtual identities, so the attacker may change the real voting result and achieve the purpose of the attack.

8.2.3. Double Spending

Double spending means that a sum of money is spent twice or more, and the attacker spends the same money twice in some way to obtain services that exceed the sum of money. Due to the reproducibility of data in the digital currency system, the system may have the same digital asset being reused due to improper operations. In 2018, there was a double-spending attack on Bitcoin Gold (BTG), which made a huge profit of more than 388,000 BTG. BTG was the 27th largest cryptocurrency in the world, with a market value of 5 billion yuan.

8.2.4. Selfish Mining

Selfish mining means that miners or mining pools withhold newly mined blocks from the public for a certain period of time and wait for the right time to broadcast all blocks to obtain greater profits. Selfish mining was proposed by Eyal et al. [95]. Compared with single-person mining, miners form a mining pool and initiate selfish mining to obtain greater profits. Driven by their own interests, rational miners are more willing to join the mining pool for selfish mining. As the size of the mining pool continues to expand, the degree of decentralization of the blockchain will decrease.

8.2.5. Replay Attack

The replay attack [96] means that the payee is dishonest, and the payee broadcasts the transaction transferred by the payer again. After some time, the transaction is written into the blockchain, and the payer thinks the transfer transaction is completed. If the payee is malicious, it rebroadcasts the transaction on the Internet, and other nodes believe it is a new transfer and deduct the payer's money twice.

8.2.6. Contract Vulnerability

The security of smart contracts is a hot topic in blockchain security. According to statistics, 89% of smart contracts have security vulnerabilities. Smart contract security issues are also more relevant as the number of smart contracts increases. According to the statistics of the blockchain security company Bcsec [97], the economic losses caused by smart contract security breaches have exceeded billions of dollars. A smart contract is a piece of code. In the design and development process, code security problems will inevitably occur, and smart contracts deployed on public chains are usually exposed to an open network environment, which further makes smart contracts vulnerable to attacks. Due to the immutability and irreversibility of the blockchain, when a hacker attacks a smart contract, the user can only watch the funds flow into the attacker's account without interrupting or preventing the execution of the contract, such as the famous DAO Attack.

DAO is a crowdfunding project initiated by the blockchain company Slock.it, which was the most prominent crowdfunding project at the time. More than 11,000 people participated in the crowdfunding of the DAO project, and the total crowdfunding value exceeded 150 million US dollars. Hackers took advantage of the loopholes in DAO's smart contract to carry out a reentrancy attack on DAO [98] and successfully stole 3.6 million ETH, more than one-third of the total ETH raised by the project. The attack eventually led to a hard fork of Ethereum, forming two chains, one for the original chain and the other for the new forked chain.

In Ethereum, smart contracts can call code from other external contracts [99]. Since smart contracts can call external contracts or send Ether, these operations require the contract to submit external calls, so attackers can use these external calls to cause attack hijacking so that the attacked contract can be re-executed at any location, bypassing the original restrictions in the code, so re-entrancy attacks occur. Re-entrancy attacks are similar to recursive calls in programming and can happen when a contract sends ETH to an unknown address.

8.3. Energy Consumption

Blockchain systems that use the PoW consensus mechanism consume a lot of energy for mining. Taking Bitcoin as an example: completing a single transaction generates 73.3 million tons of CO_2, according to the Digiconomist website [100]. It requires 1405.96 kWh of electricity, equivalent to about 48.19 days in the average U.S. household. As of October 2022, Bitcoin's estimated annual electricity consumption was 131.43 trillion kWh, leaving a yearly carbon footprint of 73.30 Mt and annual e-waste generation of 43.11 kt. The current annual electricity consumption of Bitcoin mining is comparable to the yearly electricity consumption of Argentina.

Despite the emergence of various blockchain consensus algorithms, the current blockchain consensus algorithm is still dominated by PoW, which inevitably brings energy consumption problems.

8.4. Regulation

The decentralized nature of blockchain and the automatic execution of smart contracts pose serious security regulatory challenges, leading to the proliferation of illegal acts such as money laundering and extortion. Presently, countries worldwide have very different attitudes towards blockchain technology, and the relevant research on blockchain information security regulation is still in the preliminary stage. The existing blockchain information regulation schemes are biased towards specific scenarios and have major limitations regarding functionality, efficiency, and security. Therefore, the comprehensive security regulation of blockchain information still faces great challenges and needs further research.

9. Conclusions

This paper discussed the key technologies of the blockchain around the construction of a decentralized trust mechanism. We reviewed the keys to building decentralized trust in blockchains, especially the three aspects of blockchain theory, consensus mechanisms, and smart contracts. These three aspects are also the hotspots and cores of blockchain research now, and future research will be carried out more, in combination with specific applications. After comparing the existing blockchain platforms, the architecture of the decentralized trust mechanism built by the blockchain was detailed, providing a systematic perspective for blockchain research.

Blockchains have received widespread attention due to their ability to build decentralized trust and have set off a wave of decentralization changes in various industries. Blockchain technology is booming, with emerging technologies, architectures, and applications being tried in various sectors. Although there are still many problems, blockchains are also booming. From Bitcoin to Dapps, the development of blockchain technology is not more than two decades old. The most famous blockchain application, Bitcoin, is limited to digital geeks and venture capitalists. The blockchain lacks a superstar application to become known to more people worldwide, just like Google is for search engines. Shortly, with the emergence of superstar application blockchain applications, blockchain technology will receive a new wave of research. Then, the decentralized trust mechanism will cause more profound changes in society.

Author Contributions: Conceptualization, W.D. and T.H.; investigation, W.D.; supervision, T.H.; writing—original draft, W.D.; writing—review and editing, W.D., H.W. and T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62002074, 62132018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Nakamoto, S. Bitcoin: A Peer-To-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 21 December 2022).
2. Bayer, D.; Haber, S.; Stornetta, W.S. Improving the efficiency and reliability of digital time-stamping. In *Sequences Ii*; Springer: Berlin, Germany, 1993; pp. 329–334.
3. Monrat, A.A.; Schelen, O.; Andersson, K. A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access* **2019**, *7*, 117134–117151. [CrossRef]

4. Sapra, R.; Dhaliwal, P. Blockchain: the new era of technology. In Proceedings of the 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, Waknaghat, Solan, Himachal Pradesh, India, 20–22 December 2018; pp. 495–499.
5. Buterin, V. A next-generation smart contract and decentralized application platform. *White Pap.* **2014**, *3*, 1–2.
6. Chen, Y.; Bellavitis, C. Blockchain disruption and decentralized finance: The rise of decentralized business models. *J. Bus. Ventur. Insights* **2020**, *13*, e00151. [CrossRef]
7. 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.
8. Xu, B.; Luthra, D.; Cole, Z.; Blakely, N. EOS: An architectural, performance, and economic analysis. *Retrieved June* **2018**, *11*, 2019.
9. Maesa, D.; Mori, P. Blockchain 3.0 applications survey. *J. Parallel Distrib. Comput.* **2020**, *138*, 99–114. [CrossRef]
10. Alammary, A.; Alhazmi, S.; Almasri, M.; Gillani, S. Blockchain-based applications in education: A systematic review. *Appl. Sci.* **2019**, *9*, 2400. [CrossRef]
11. Agbo, C.C.; Mahmoud, Q.H.; Eklund, J.M. Blockchain technology in healthcare: A systematic review. *Healthcare* **2019**, *7*, 56. [CrossRef]
12. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
13. Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *88*, 173–190. [CrossRef]
14. Alabdulwahhab, F.A. Web 3.0: The Decentralized Web Blockchain networks and Protocol Innovation. In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), IEEE, Hangzhou, China, 24–27 October 2018; pp. 1–4.
15. Xu, M.; Chen, X.; Kou, G. A systematic review of blockchain. *Financ. Innov.* **2019**, *5*, 1–14. [CrossRef]
16. Yli-Huumo, J.; Ko, D.; Choi, S.; Park, S.; Smolander, K. Where is current research on blockchain technology?—A systematic review. *PLoS ONE* **2016**, *11*, e0163477. [CrossRef] [PubMed]
17. Paulavičius, R.; Grigaitis, S.; Igumenov, A.; Filatovas, E. A Decade of Blockchain: Review of the Current Status, Challenges, and Future Directions. *Informatica* **2019**, *30*, 729–748. [CrossRef]
18. Zheng, Z.; Xie, S. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
19. Sambra, A.V.; Mansour, E.; Hawke, S.; Zereba, M.; Greco, N.; Ghanem, A.; Zagidulin, D.; Aboulnaga, A.; Berners-Lee, T. *Solid: A Platform for Decentralized Social Applications Based on Linked Data*; Technical Report; MIT CSAIL & Qatar Computing Research Institute: Asheville, NC, USA, 2016.
20. Raman, A.; Joglekar, S.; De Cristofaro, E.; Sastry, N.; Tyson, G. Challenges in the Decentralized Web: The Mastodon Case. In Proceedings of the 19th ACM Internet Measurement Conference (IMC 2019). ACM, Amsterdam, Netherlands, 21–31 October 2019; pp. 217–229.
21. Seong, S.W.; Seo, J.; Nasielski, M.; Sengupta, D.; Hangal, S.; Teh, S.K.; Chu, R.; Dodson, B.; Lam, M.S. Prpl: a decentralized social networking infrastructure. In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, San Francisco, CA, USA, 15 June 2010; pp. 1–8.
22. Bielenberg, A.; Helm, L.; Gentilucci, A.; Stefanescu, D.; Zhang, H. The growth of diaspora-a decentralized online social network in the wild. In Proceedings of the 2012 IEEE INFOCOM workshops, IEEE, Orlando, FL, USA, 25–30 March 2012; pp. 13–18.
23. Dang, H.; Dinh, T.T.A.; Loghin, D.; Chang, E.C.; Lin, Q.; Ooi, B.C. Towards scaling blockchain systems via sharding. In Proceedings of the 2019 International Conference on Management of Data, Amsterdam, The Netherlands, 30 June–5 July 2019; pp. 123–140.
24. Graf, M.; Rausch, D.; Ronge, V.; Egger, C.; Küsters, R.; Schröder, D. A security framework for distributed ledgers. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, 15–19 November 2021; pp. 1043–1064.
25. Xu, C.; Zhang, C.; Xu, J. vchain: Enabling verifiable boolean range queries over blockchain databases. In Proceedings of the 2019 International Conference on Management of Data, Amsterdam, The Netherlands, 12–19 July 2019; pp. 141–158.
26. Yu, H.; Nikolić, I.; Hou, R.; Saxena, P. Ohie: Blockchain scaling made simple. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), IEEE, Hyatt Regency, San Francisco, CA, USA, 17–21 May 2020; pp. 90–105.
27. Ruan, P.; Kanza, Y.; Ooi, B.C.; Srivastava, D. LedgerView: Access-Control Views on Hyperledger Fabric. In Proceedings of the 2022 International Conference on Management of Data; Association for Computing Machinery, New York, NY, USA, 14–19 June 2022; pp. 2218–2231. [CrossRef]
28. Han, Y.; Li, C.; Li, P.; Wu, M.; Zhou, D.; Long, F. Shrec: Bandwidth-Efficient Transaction Relay in High-Throughput Blockchain Systems. In Proceedings of the 11th ACM Symposium on Cloud Computing; Association for Computing Machinery, New York, NY, USA, 19–21 October 2020; pp. 238–252. [CrossRef]
29. Sestrem Ochôa, I.; Augusto Silva, L.; De Mello, G.; Garcia, N.M.; de Paz Santana, J.F.; Quietinho Leithardt, V.R. A cost analysis of implementing a blockchain architecture in a smart grid scenario using sidechains. *Sensors* **2020**, *20*, 843. [CrossRef] [PubMed]
30. Al-Mamun, A.; Yan, F.; Zhao, D. SciChain: Blockchain-enabled Lightweight and Efficient Data Provenance for Reproducible Scientific Computing. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Crete, Greece, 19–23 April 2021; pp. 1853–1858. [CrossRef]

31. Kaur, M.; Gupta, S.; Kumar, D.; Verma, C.; Neagu, B.C.; Raboaca, M.S. Delegated Proof of Accessibility (DPoAC): A Novel Consensus Protocol for Blockchain Systems. *Mathematics* **2022**, *10*, 2336. [CrossRef]
32. Lansky, J.; Rahmani, A.M.; Ali, S.; Bagheri, N.; Safkhani, M.; Hassan Ahmed, O.; Hosseinzadeh, M. BCmECC: A Lightweight Blockchain-Based Authentication and Key Agreement Protocol for Internet of Things. *Mathematics* **2021**, *9*, 3241. [CrossRef]
33. AbuDaqa, A.A.; Mahmoud, A.; Abu-Amara, M.; Sheltami, T. Survey of network coding based P2P file sharing in large scale networks. *Appl. Sci.* **2020**, *10*, 2206. [CrossRef]
34. Milojicic, D.S.; Kalogeraki, V.; Lukose, R.; Nagaraja, K.; Pruyne, J.; Richard, B.; Rollins, S.; Xu, Z. Peer-to-Peer Computing. 2002. Available online: https://www.hpl.hp.com/techreports/2002/HPL-2002-57R1.pdf (accessed on 18 November 2022).
35. Aljaedi, A.R.A. Peer-to-Peer Network Security Issues and Analysis: Review. *Int. J. Comput. Sci. Netw. Secur.* **2020**, *20*, 74–88.
36. Risson, J.; Moors, T. Survey of Research Towards Robust Peer-to-Peer Networks: Search methods. *Comput. Netw.* **2007**, *50*, 3485–3521. [CrossRef]
37. Fischer, M.J.; Lynch, N.A.; Paterson, M.S. Impossibility of distributed consensus with one faulty process. *J. ACM* **1985**, *32*, 374–382. [CrossRef]
38. Fox, A.; Brewer, E.A. Harvest, yield, and scalable tolerant systems. In Proceedings of the Seventh Workshop on Hot Topics in Operating Systems, Rio Rico, AZ, USA, 29–30 March 1999; pp. 174–178.
39. Qi, Z.; Zhang, Y.; Wang, Y.; Wang, J.; Wu, Y. A cascade structure for blockchain. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15 August 2018; pp. 252–253.
40. Ribero, Y.; Raissar, D. DagCoin Whitepaper. 2018. Available online: https://prismic-io.s3.amazonaws.com/dagcoin/f4e531e1-a5db-43b6-930c-14bf705e65ee_Dagcoin_White_Paper.pdf (accessed on 27 October 2022).
41. Zhang, Y.; Xu, C.; Li, H.; Yang, H.; Shen, X. Chronos: Secure and accurate time-stamping scheme for digital files via blockchain. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
42. Ma, G.; Ge, C.; Zhou, L. Achieving reliable timestamp in the bitcoin platform. *Peer Peer Netw. Appl.* **2020**, *13*, 2251–2259. [CrossRef]
43. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
44. Mingxiao, D.; Xiaofeng, M.; Zhe, Z.; Xiangwei, W.; Qijun, C. A review on consensus algorithm of blockchain. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff Center, Banff, AB, Canada, 5—8 October 2017; pp. 2567–2572. [CrossRef]
45. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.* **2002**, *20*, 398–461. . [CrossRef]
46. Wu, Y.; Song, P.; Wang, F. Hybrid consensus algorithm optimization: A mathematical method based on POS and PBFT and its application in blockchain. *Math. Probl. Eng.* **2020**, *2020*, 13. [CrossRef]
47. Biswas, S.; Sharif, K.; Li, F.; Maharjan, S.; Mohanty, S.P.; Wang, Y. PoBT: A lightweight consensus algorithm for scalable IoT business blockchain. *IEEE Internet Things J.* **2019**, *7*, 2343–2355. [CrossRef]
48. Fu, X.; Wang, H.; Shi, P. A survey of Blockchain consensus algorithms: mechanism, design and applications. *Sci. China Inf. Sci.* **2021**, *64*, 1–15. [CrossRef]
49. Dwork, C.; Naor, M. Pricing via Processing or Combatting Junk Mail. In Proceedings of the International Cryptology Conference on Advances in Cryptology, Heidelberg, Berlin, 22–26 August 1993.
50. Raghav; Andola, N.; Venkatesan, S.; Verma, S. PoEWAL: A lightweight consensus mechanism for blockchain in IoT. *Pervasive Mob. Comput.* **2020**, *69*, 101291. [CrossRef]
51. Huang, J.; Kong, L.; Chen, G.; Wu, M.Y.; Liu, X.; Zeng, P. Towards Secure Industrial IoT: Blockchain System with Credit-Based Consensus Mechanism. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3680–3689. [CrossRef]
52. Ethereum. 2022. Available online: https://ethereum.org/zh/upgrades/beacon-chain/ (accessed on 4 October 2022).
53. Snider, M.; Samani, K.; Jain, T. Delegated Proof of Stake: Features & Tradeoffs. 2018. Available online: https://holbrook.no/share/papers/DPoS_-Features-and-Tradeoffs.pdf (accessed on 10 October 2022).
54. Luo, Y.; Chen, Y.; Chen, Q.; Liang, Q. A new election algorithm for DPos consensus mechanism in blockchain. In Proceedings of the 2018 7th International Conference on Digital Home (ICDH), Sfax, Tunisia, 15–16 October 2018; pp. 116–120.
55. Saad, M.; Qin, Z.; Ren, K.; Nyang, D.; Mohaisen, D. e-pos: Making proof-of-stake decentralized and fair. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1961–1973. [CrossRef]
56. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; pp. 357–388.
57. Abraham, I.; Gueta, G.; Malkhi, D. Hot-Stuff the Linear, Optimal-Resilience, One-Message BFT Devil. *CoRR* 2018, abs/1803.05069.
58. Teng, L. Improvement and Application of Practical Byzantine Fault Tolerance Consensus Algorithm. Ph.D. Thesis, Hebei University of Engineering, Handan, China, 2021.
59. YUAN Yong, W.F.Y. Blockchain: The State of the Art and Future Trends. *Acta Autom. Sin.* **2016**, *42*, 481–494.
60. Montes, J.M.; Ramirez, C.E.; Gutierrez, M.C.; Larios, V.M. Smart Contracts for supply chain applicable to Smart Cities daily operations. In Proceedings of the 2019 IEEE International Smart Cities Conference (ISC2), Virtual, 16–18 November 2019.

61. Vujičić, D.; Jagodić, D.; Ranđić, S. Blockchain technology, bitcoin, and Ethereum: A brief overview. In Proceedings of the 2018 17th International Symposium Infoteh-Jahorina (Infoteh), Jahorina, Bosnia and Herzegovina, 21–23 March 2018; pp. 1–6.
62. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
63. Kalra, S.; Goel, S.; Dhawan, M.; Sharma, S. Zeus: analyzing safety of smart contracts. In Proceedings of the 25th Annual Network and Distributed System Security Symposium, San Diego, CA, USA, 18–21 February 2018; pp. 1–12.
64. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
65. Wohrer, M.; Zdun, U. Smart contracts: security patterns in the ethereum ecosystem and solidity. In Proceedings of the 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 20 March 2018; pp. 2–8.
66. Kaleem, M.; Mavridou, A.; Laszka, A. Vyper: A security comparison with solidity based on common vulnerabilities. In Proceedings of the 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Virtual, 28–30 September 2020; pp. 107–111.
67. Buterin, V. Vyper Documentation. *Retrieved Oct.* **2018**, *30*, 2018.
68. Kfir, S.; Fournier, C. DAML: The contract language of distributed ledgers. *Commun. ACM* **2019**, *62*, 48–54. [CrossRef]
69. Back, A. Hashcash-A Denial of Service Counter-Measure. 2002. Available online: https://www.researchgate.net/publication/2482110_Hashcash_-_A_Denial_of_Service_Counter-Measure (accessed on 1 April 2022).
70. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3154–3164. [CrossRef]
71. Su, Z.; Wang, Y.; Xu, Q.; Fei, M.; Tian, Y.C.; Zhang, N. A secure charging scheme for electric vehicles with smart communities in energy blockchain. *IEEE Internet Things J.* **2018**, *6*, 4601–4613. [CrossRef]
72. Kuo, T.T.; Kim, H.E.; Ohno–Machado, L. Blockchain distributed ledger technologies for biomedical and health care applications. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 1211–1220. [CrossRef]
73. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. Medrec: Using blockchain for medical data access and permission management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 25–30.
74. Mettler, M. Blockchain technology in healthcare: The revolution starts here. In Proceedings of the 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), Munich, Germany, 14–17 September 2016; pp. 1–3.
75. Hirtan, L.; Krawiec, P.; Dobre, C.; Batalla, J.M. Blockchain-based approach for e-health data access management with privacy protection. In Proceedings of the 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11–13 September; pp. 1–7.
76. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
77. Alphand, O.; Amoretti, M.; Claeys, T.; Dall'Asta, S.; Duda, A.; Ferrari, G.; Rousseau, F.; Tourancheau, B.; Veltri, L.; Zanichelli, F. IoTChain: A blockchain security architecture for the Internet of Things. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 16–18 April 2018; pp. 1–6.
78. Li, C.; Zhang, L.J. A blockchain based new secure multi-layer network model for internet of things. In Proceedings of the 2017 IEEE International Congress on Internet of Things (ICIOT), Honolulu, HI, USA, 25–30 June 2017; pp. 33–41.
79. Pinno, O.J.A.; Gregio, A.R.A.; De Bona, L.C. Controlchain: Blockchain as a central enabler for access control authorizations in the iot. In Proceedings of the GLOBECOM 2017-2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
80. Yeom, S.; Giacomelli, I.; Fredrikson, M.; Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF), Oxford, UK, 9–12 July 2018; pp. 268–282.
81. Zyskind, G.; Nathan, O.; Pentland, A. Decentralizing privacy: Using blockchain to protect personal data. In Proceedings of the 2015 IEEE Security and Privacy Workshops, Fairmont, San Jose, CA, USA, 18–20 May 2015; pp. 180–184.
82. Chen, X.; Ji, J.; Luo, C.; Liao, W.; Li, P. When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 1178–1187.
83. Qi, Y.; Hossain, M.S.; Nie, J.; Li, X. Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Future Gener. Comput. Syst.* **2021**, *117*, 328–337. [CrossRef]
84. Fairfield, J.A. Tokenized: The law of non-fungible tokens and unique digital property. *Ind. LJ* **2022**, *97*, 1261.
85. Wang, Q.; Li, R.; Wang, Q.; Chen, S. Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges. *arXiv* **2021**, arXiv:2105.07447. https://doi.org/10.48550/ARXIV.2105.07447.
86. Bambacht, J.; Pouwelse, J. Web3: A Decentralized Societal Infrastructure for Identity, Trust, Money, and Data. *arXiv* **2022**, arXiv:2203.00398.
87. Momtaz, P.P. Some very simple economics of web3 and the metaverse. *FinTech* **2022**, *1*, 225–234. [CrossRef]
88. Ahn, G.J.; Shehab, M.; Squicciarini, A. Security and privacy in social networks. *IEEE Internet Comput.* **2011**, *15*, 10–12. [CrossRef]

89. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A fistful of bitcoins: characterizing payments among men with no names. In Proceedings of the 2013 Conference on Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013; pp. 127–140.
90. Al-Haija, Q.A.; Alsulami, A.A. High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks. *Electronics* **2021**, *10*, 2113. [CrossRef]
91. Javaid, H.; Yang, J.; Santoso, N.; Upadhyay, M.; Mohan, S.; Hu, C.; Brebner, G. Blockchain machine: A network-attached hardware accelerator for hyperledger fabric. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), Bologna, Italy, 10–13 July 2022; pp. 258–268.
92. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. Available online: https://coinrivet.com/research/papers/the-bitcoin-lightning-network-scalable-off-chain-instant-payments/ (accessed on 6 October 2022).
93. Back, A.; Corallo, M.; Dashjr, L.; Friedenbach, M.; Maxwell, G.; Miller, A.; Poelstra, A.; Timón, J.; Wuille, P. Enabling Blockchain Innovations with Pegged Sidechains. 2014, Volume 72, pp. 201–224. Available online: http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains (accessed on 6 October 2022).
94. Yang, H.; Zhong, Y.; Yang, B.; Yang, Y.; Xu, Z.; Wang, L.; Zhang, Y. An Overview of Sybil Attack Detection Mechanisms in VFC. In Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Baltimore, America, 27–30 June 2022; pp. 117–122. [CrossRef]
95. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM* **2018**, *61*, 95–102. [CrossRef]
96. Zhang, R.; Xue, R.; Liu, L. Security and Privacy on Blockchain. *ACM Comput. Surv.* **2019**, *52*, 1–34. [CrossRef]
97. Bcsec. 2022. Available online: https://hacked.slowmist.io/ (accessed on 10 October 2022).
98. Atzei, N.; Bartoletti, M.; Cimoli, T. A Survey of Attacks on Ethereum Smart Contracts. Cryptology ePrint Archive, Paper 2016/1007, 2016. Available online: https://eprint.iacr.org/2016/1007 (accessed on 14 October 2022).
99. Chen, H.; Pendleton, M.; Njilla, L.; Xu, S. A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses. *ACM Comput. Surv.* **2020**, *53*, 1–43. [CrossRef]
100. Digiconomist. 2022. Available online: https://digiconomist.net/ (accessed on 14 October 2022).