Review

Blockchain and Distributed Systems: Basic Concepts and Implications

Mauricio Witter * and Antonio Rodrigo De Vit

Posted Date: 26 March 2024

doi: 10.20944/preprints202403.1467.v1

Keywords: Blockchain; Peer-to-Peer; RPC; Overlay Network

 doi:10.20944/preprints202403.1467.v1

Article

Blockchain and Distributed Systems: Basic Concepts and Implications

Maurício Witter [1,*] and A. Rodrigo De Vit [2]

[1] UFSM, Constantina, BRA
[2] UFSM, Frederico Westphalen, BRA; rodrigodevit@inf.ufsm.br
* Correspondence: mauricio.witter@ufsm.mail.com

Abstract: Blockchain technology has emerged as a necessity for the decentralization of payment methods and transactions, but it has brought with it many properties of distributed systems that have made it a crucial technology for overcoming some of society's challenges, especially in the context of decentralizing services, transparency of information, availability, and security. Its architecture and communication methods, although possessing some complex nuances to understand, particularly for the lay audience in the field of distributed systems, protocols, and computer networks. In this article, we will explore some topics of distributed systems related to blockchain technology.

Keywords: Blockchain; Peer-to-Peer; RPC

1. Introduction

Blockchain technology emerged as a disruptive technology that quickly gained attention, mainly for its prominent application in digital assets, soon known as Bitcoin. The term "Blockchain" is used to describe a data structure, sometimes the system as a whole [p. 244] [1], but for this context, we define it as a data structure where "Block" refers to a block of transactions and "chain" refers to the chain that connects the blocks through a hash. Thus, Blockchain is an ordered and chained sequence of blocks, where the subsequent block contains a hash of the representation of the previous block, as depicted in Figure 3.

Blockchain technology makes use of a Peer-to-Peer network [p. 243] [1], which is defined as an overlay network. Peer-to-peer (P2P) networks are distributed systems by nature, without any hierarchical organization or centralized control. Peers form self-organized virtual network topologies over the physical network topology [p. 1] [2]. Essentially, network nodes form a virtual network that uses general protocols to operate over the Internet Protocol (IP) and thus connect peers in the network.

In the context of software architecture, Blockchain technology enables new forms of distributed software architectures, where agreement on shared state for decentralized and transactional data can be established through a large network of untrusted participants [p. 243] [1]. In other words, there is no establishment of trust between any of the parties; the network node can be a well-intentioned individual, an entity, or a malicious individual, as there is no establishment of prior trust connections since the consensus algorithm ensures transaction validity and security, which is crucial for decentralized scenarios.

This article aims to analyze the architecture of Blockchain technology and how it incorporates concepts from distributed systems to make applications robust, resilient, fault-tolerant, decentralized, and secure. This work can be used in the future to develop practical applications or serve as a guide to understanding some of the concepts of blockchain technologies.

2. Peer-to-Peer

Unlike the client-server model, which communicates directly with the TCP/IP model, Peer-to-Peer networks are virtual networks implemented above the TCP/IP model. Figure 1 provides a visual conceptualization of how the overlay network interacts with the physical network. For example, if overlay node **A** has traffic to node **D**, it can route it directly using the tunnel from **A** to **D** or relay it through another overlay node **B** or **C** [3].

 doi:10.20944/preprints202403.1467.v1

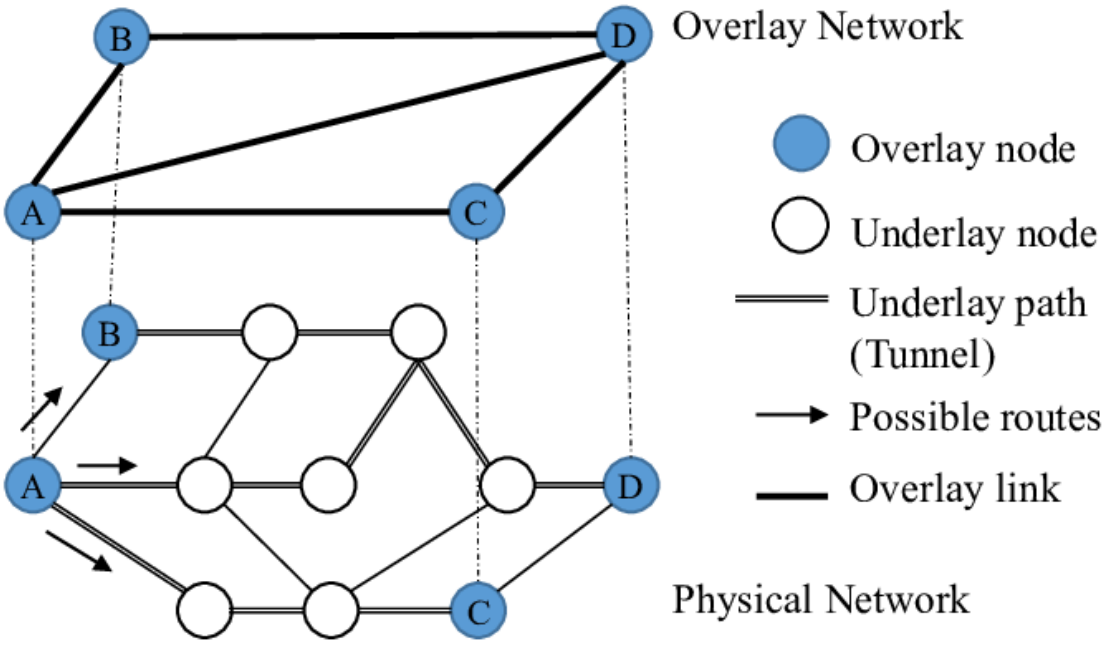

Figure 1. Overlay network architecture [3].

It is important to note that the actual communication between two nodes connected by a link in the overlay is carried out through connections that exist in one or more underlying network layers. The physical network still handles the basic mechanisms of transport, addressing, and routing. Higher-level services like overlay networks use these mechanisms as a means to the internet [p. 17–18] [4].

While the physical network handles routing and transport for the internet, custom addressing and routing schemes can be implemented for the nodes in the higher level, which is the case in most Peer-to-Peer applications, such as Blockchain. Thus, high-level overlays can be used to develop new services and applications without the need to deploy new devices or protocols at the base [p. 18] [4].

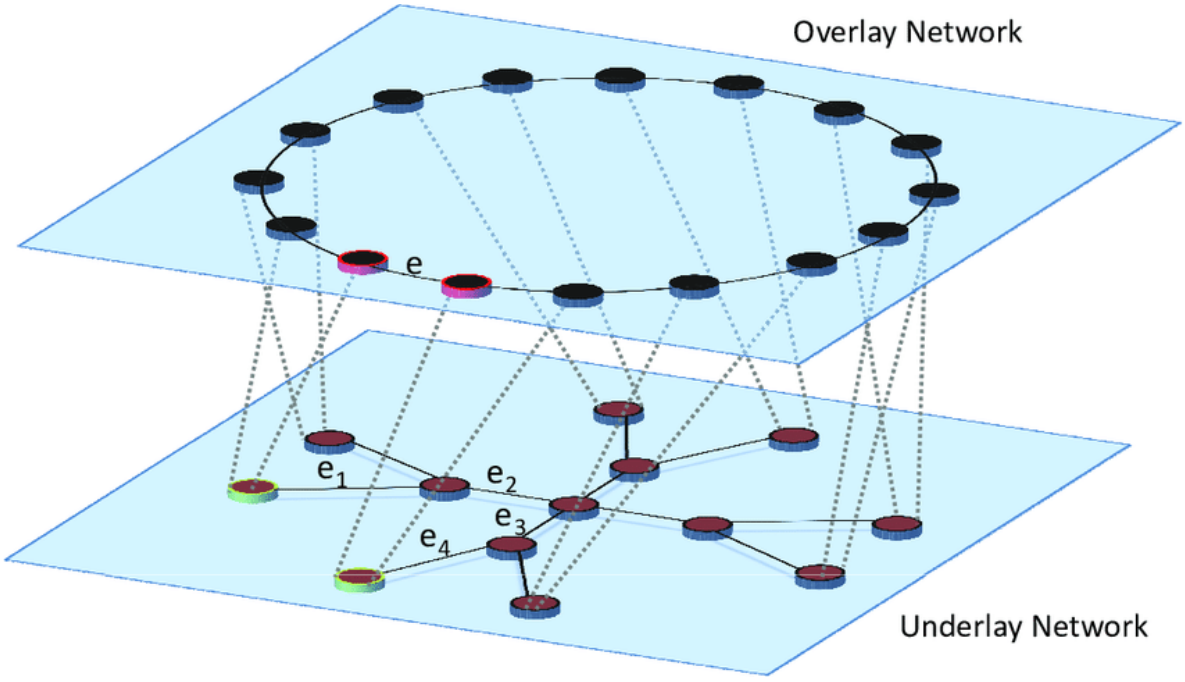

Figure 2. Overlay Topology (above) and Underlay Topology (below) [4].

 doi:10.20944/preprints202403.1467.v1

As stated by [p. 1] [2], overlay networks possess a combination of various features such as robust wide-area routing architecture, efficient data item lookup, selection of nearby peers, redundant storage, persistence, hierarchical naming, trust and authentication, anonymity, massive scalability, and fault tolerance.

Although it may seem so, peer selection is not simply random. According to [p. 72–73] [2], the topology of the P2P overlay network is tightly controlled, and content is placed in specific locations to make subsequent queries more efficient. Such structured P2P systems use Distributed Hash Table (DHT), in which the location information of the data object (or value) is deterministically placed on peers with identifiers corresponding to the unique key of the data object.

Systems based on DHT have a property that consistently assigns random uniform NodeIDs to the set of peers in a large identifier space. Each peer then maintains a small routing table consisting of NodeIDs and IP addresses of its neighboring peers. Search queries or message routing are forwarded through overlay paths to peers progressively, with NodeIDs that are closest to the key in the identifier space [p. 73] [2].

According to [p. 76-79] [5], Ethereum and the InterPlanetary File System (IPFS) make use of the Kademlia DHT algorithm. Kademlia is a Distributed Hash Table (DHT) algorithm used in peer-to-peer (P2P) networks. This algorithm is primarily used to maintain a list of nodes in a P2P network efficiently and scalably. It is known for its ability to search for nodes and data in the network quickly and effectively. It uses node IDs and resource IDs to organize nodes into a binary tree structure.

Each node in the Kademlia network is identified by a unique ID, which is a fixed-size binary sequence, usually 160 bits. Nodes are organized into a binary tree, and the proximity between two nodes is calculated using the XOR distance between their IDs. The closer the IDs, the closer the nodes are in the tree [p. 79–80] [2].

Thus, the use of overlay networks has enabled new ways of creating distributed systems, and new technologies like Blockchain allow for the creation of a collaborative and highly scalable ecosystem. Peers share their computational resources and, in addition to being clients, they become servers. In other words, each peer in the network acts bidirectionally, performing both roles, which eliminates the need for centralized servers and hiring cloud computing services.

It is worth noting that the implementation of overlay networks is highly complex since there is no RFC (Request for Comments) or international standardization for implementation. Addressing and routing models can be customized in the overlay network, granting it freedom but also introducing complexity.

3. Blockchain

Blockchain as a distributed system has many elements, as previously mentioned, it operates on an overlay network and utilizes the processing power, bandwidth, and storage capacity of network peers' machines. Moreover, it has chains and blocks to securely store data. Nonetheless, Blockchain is a construct of cryptographic algorithms, consensus, and code execution (Smart Contracts).

The first generation of Blockchain came with Bitcoin [6]. This elementary generation had many limitations that have been improved over time. But especially, the second generation of Blockchain brought a revolution by allowing users to deploy and execute code on a Blockchain (Turing-complete) [p. 244] [1].

A Blockchain implements a distributed ledger, which can generally verify and store any type of transaction ([p. 1–2] [1] as cited in 7). Each node in the network has an identical copy of the ledger. Consequently, network nodes can verify transactions transparently and reliably through consensus algorithms, eliminating the need for a trusted central server.

The ledger is the database of an elementary Blockchain, structured by the chain of blocks, as shown in Figure 3. This ledger records all transactions made by the peers of the Blockchain network and is public.

 doi:10.20944/preprints202403.1467.v1

Figure 3 depicts a representation of how the blockchains' block chains are. Essentially, there is a linear chain of blocks, where each block has its hash and the hash of the previous content.

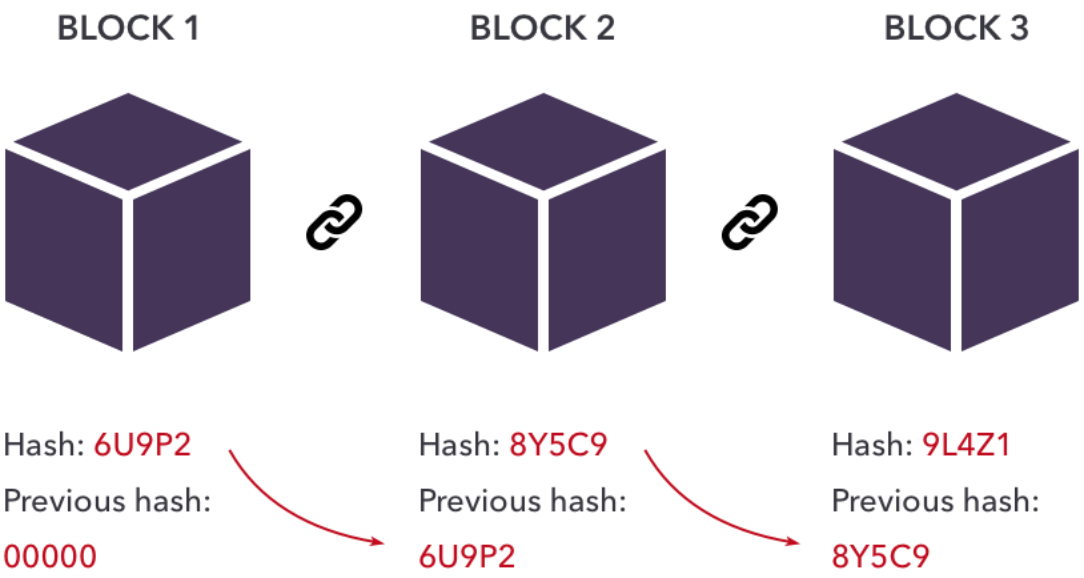

Figure 3. Data structure used to represent Blockchain blocks and chains [8].

Transactions consist of a payload, in other words, data packets that store parameters such as monetary value, recipient address, and results of function calls (such as smart contracts) [1].

A transaction goes through several steps before being actually recorded in the ledger, as shown in Figure 4. According to [1], a transaction is signed by its initiator to authorize the payload data of a transaction or the creation and execution of a smart contract.

The sender signs the transaction with their private key to prove ownership of the transaction. The transaction is propagated to the nodes connected to the Blockchain network, which perform initial validation to check if the transaction complies with protocol rules. When the majority of peers agree that the transaction is valid, it is sent to the pool of valid transactions (mempool).

Network miners (mining nodes) mine blocks using the Proof-of-Work (PoW) algorithm. When a node creates a block, this block is propagated to other nodes in the network, where it also undergoes validation requiring consensus among the majority of peers that it is a valid block [1, p. 244]. Thus, when the block is considered valid by peers, it is added to their copy of the Blockchain and propagated to other nodes in the network.

Transactions that were in the mempool are chosen by miners, typically based on the fee paid by the sender. In other words, the higher the fee, the faster the transaction will be confirmed by miners. Transactions are collected from the mempool and inserted into valid blocks, which are then propagated to network peers. Ultimately, transactions are recorded immutably and permanently in the Blockchain. As stated by [1], consensus ensures that all stored transactions are valid and that each valid transaction is added only once.

 doi:10.20944/preprints202403.1467.v1

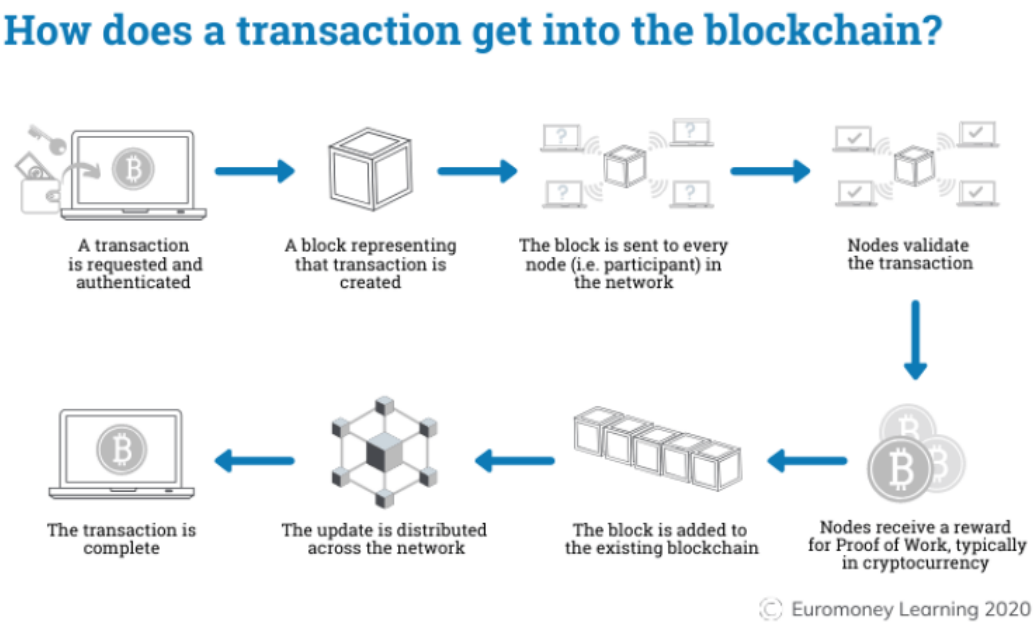

Figure 4. Steps a transaction goes through until it is considered valid and recorded in the ledger [9].

4. Security

Security is another crucial point in Distributed Systems. Blockchain implements many cryptographic algorithms to ensure that transactions are secure. Much of the security is provided by consensus algorithms, where all transactions and blocks need to be validated by the majority of nodes in the blockchain network. According to [1], the integrity of a transaction is verified by algorithmic rules and cryptographic techniques.

The transaction history in the Blockchain cannot be deleted or altered without invalidating the entire chain of hashes. Combined with computational constraints and block creation incentive schemes, this prevents tampering and revision of the information stored in the Blockchain [p. 244] [1].

Public key cryptography and digital signatures are typically used to identify accounts and ensure the authorization of transactions initiated on a blockchain [p. 244] [1].

There are also other techniques used in Distributed Systems that are employed by blockchains, such as Byzantine fault tolerance (BFT), an alternative to the PoW consensus algorithm of [6]. According to [p. 250] [1], BFT requires all participants to agree on the network's participant list, so it is usually used for private blockchains. It is a more conventional approach in distributed systems and offers much stronger consistency guarantees and lower latency but for a smaller number of participants. According to [p. 250] [1], BFT ensures consensus despite the arbitrary behavior of some fraction of participants.

5. Consensus

The choice of consensus protocol impacts security and scalability. As soon as a new block is generated by a miner, the miner propagates the block to its connected peers in the blockchain network. However, miners may encounter different competing new blocks and resolve this using the blockchain's consensus mechanisms [1]. The fundamental approach proposed by [p. 1] [6] was the Proof-of-Work (PoW) consensus algorithm.

In Bitcoin, which uses the PoW algorithm, new blocks are generated through the Proof-of-Work mechanism. Bitcoin miners compete with each other to solve simple but time-consuming mathematical calculations to decompose, which is done for each block, using large amounts of computational energy

 doi:10.20944/preprints202403.1467.v1

[p. 248] [1]. Blocks, like transactions, need to pass through the consensus and approval of the majority of peers. In block mining, miners compete to generate blocks, and blocks compete to become part of one of the blockchains. In Bitcoin, the consensus principle is that the longest chain is chosen, and the other chains are abandoned by the nodes, with only one being validated. Ultimately, the miner who generated the winning block receives their mining reward plus the fees paid by other users to validate transactions.

Decentralized systems using anonymous validators need protection against Sybil attacks, where attackers create many hostile anonymous nodes. Bitcoin partially protects against this by using its Proof-of-Work mechanism, so it is not the total number of nodes that is important for integrity, but rather the total amount of computational power. Although it is easy for an attacker to create anonymous nodes, it is not easy for them to accumulate large amounts of computational power [p. 245] [1] (as cited in [10]).

6. Smart Contracts

Smart Contracts are a way to program contracts (algorithms) by anyone with the intention of executing them on a Turing-complete blockchain [p. 248] [1]. According to [p. 116675] [11], a smart contract is defined as a computer program that enforces the agreed-upon promises by the interacting parties in the absence of trusted intermediaries. Thus, computation in a blockchain-based system can be performed on-chain, for example, through smart contracts, or off-chain.

With the development of the Ethereum ecosystem, the smart contract becomes a central point for leveraging blockchains as programmable state machines, introducing the execution of decentralized applications (dApps) [p. 116675] [11].

Since the introduction of smart contracts, blockchain applications are no longer limited to creating and managing tokens and digital assets; various platforms with smart contract capabilities have emerged to connect blockchains [p. 116673] [11].

Smart contracts can be developed using the Solidity programming language. This is a Turing-complete and Object-Oriented language developed by the Ethereum platform to execute smart contracts on the Ethereum Virtual Machine (EVM) [p. 116673] [11].

You don't need to be a node in the network to interact with the blockchain. Instead, when a user adds a new entry to the ledger of a blockchain, they send a transaction to an existing node using a Remote Procedure Call (RPC) protocol. This node then relays the transaction to the rest of the network for inclusion in a future block. This means that the parties involved in a transaction, such as the sender and the recipient, are not directly involved in executing that transaction. Instead, this task falls to the network peers, who confirm and validate the transaction [p. 4] [12]. Thus, the network nodes provide a JSON-RPC interface that allows any client to interact with the blockchain through this communication interface.

Off-chain data storage can be a private cloud in the client's infrastructure or a public storage provided by third parties or a network. Some peer-to-peer data storages are designed to be compatible with blockchain, such as IPFS and Storj [p. 248] [1].

One challenge of smart contracts is that once deployed to the main network of a blockchain, it cannot be changed anymore, due to the fundamental principle of immutability of blockchains.

7. Transparency

Transparency is a crucial factor in distributed systems. Blockchain, as one of these systems, possesses several transparency properties, such as access transparency, where participants have access to the data and transactions recorded on the blockchain through standardized operations.

Concurrency transparency in consensus algorithms like PoW by [6] and PoS allows simultaneous transactions without interference between them.

Peer-to-peer network is a form of replication transparency, where each node in the network holds a copy of the ledger, and transactions, block insertions, and chains are relayed to network nodes

 doi:10.20944/preprints202403.1467.v1

constantly to ensure everyone has the same copy, attributes that ensure availability and fault tolerance, thus end-users need not worry about data replication.

Failure transparency is one of the primary issues with centralized servers that blockchain and peer-to-peer networks resolve, as even if some nodes in the network fail due to a hypothetical catastrophe, there are many other active nodes, preventing network unavailability.

It also addresses mobility transparency; network nodes can connect and disconnect seamlessly, without any apparent impact on the peer-to-peer network.

8. Conclusion

Peer-to-peer networks have become increasingly popular in recent years, primarily due to the need to decentralize the web as a whole. Centralized services, while still prevalent, have several disadvantages, such as centralized servers, which entail availability complexities. In case a server at the central point fails and there are no means for fault recovery, the service goes offline.

Furthermore, as stated by [1], in a centralized system, all users depend on a central authority to mediate transactions. Peer-to-peer networks, however, do not suffer from this problem, as they utilize network node resources and can remain available at all times, with transactions being autonomous and consensus-based. For example, transactions on blockchain networks can be conducted at any time, whether on holidays or weekends, given their automation. In contrast, non-autonomous centralized services go offline during occasional or routine periods.

Despite all these benefits, there are some limitations, such as real-time systems, data size on the blockchain, transaction confirmation latency, and costs [1]. First-generation blockchains like Bitcoin and Ethereum have high transaction costs, and validations can take seconds to minutes. Additionally, the larger the transaction payload size (MB/s), the higher the fee charged for nodes to validate. A common practice for managing data in blockchain-based systems is to store raw data off-chain and store only metadata, small critical data, and hashes on-chain [p. 247] [1].

Some next-generation blockchains like Solana and Fantom employ alternative Proof-of-Stake (PoS) algorithms and Directed Acyclic Graph (DAG) data structures to organize and validate transactions more efficiently without sacrificing security.

In this article, we discussed how concepts from distributed systems are applied to real-world systems to provide highly available, distributed applications through overlay networks and cryptographic algorithms to provide a secure system for applications, where the challenge lies in trusting peers without prior knowledge of goodwill.

Future work could delve deeper into communication protocols between overlay networks, physical networks, and also explore third-generation blockchains with new ledger structures and consensus algorithms, where interoperability between various blockchains and decentralized governance is achieved.

References

1. Xu, X.; Weber, I.; Staples, M.; Zhu, L.; Bosch, J.; Bass, L.; Pautasso, C.; Rimba, P. A Taxonomy of Blockchain-Based Systems for Architecture Design. *2017 IEEE International Conference on Software Architecture (ICSA)* **2017**, pp. 243–252. doi:10.1109/ICSA.2017.33.
2. Lua, E.K.; Crowcroft, J.; Pias, M.; Sharma, R.; Lim, S. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials* **2005**, *7*, 72–93. doi:10.1109/COMST.2005.1610546.
3. Rai, A.; Singh, R.; Modiano, E. A Distributed Algorithm for Throughput Optimal Routing in Overlay Networks **2016**.
4. Scholtes, I. Harnessing Complex Structures and Collective Dynamics in Large Networked Computing Systems **2011**. pp. 17–18.
5. Eisenbarth, J.P.; Cholez, T.; Perrin, O. Avoiding the 1 TB Storage Wall: Leveraging Ethereum's DHT to Reduce Peer Storage Needs. Proceedings of the 5th ACM International Symposium on Blockchain and

Secure Critical Infrastructure; Association for Computing Machinery: New York, NY, USA, 2023; BSCI '23, p. 75–84. doi:10.1145/3594556.3594625.

6. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System **2009**. pp. 1–9.
7. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys & Tutorials* **2016**, *18*, 2084–2123. doi:10.1109/COMST.2016.2535718.
8. Dimitriadis, V.; Maglaras, L.; Polemi, N.; Kantzavelou, I.; Ayres, N. Uncuffed: A Blockchain-Based Secure Messaging System. Proceedings of the 25th Pan-Hellenic Conference on Informatics; Association for Computing Machinery: New York, NY, USA, 2022; PCI '21, p. 340–345. doi:10.1145/3503823.3503886.
9. Singh, S.; Chakraverty, S. Implementation of Proof-of-Work using Ganache. 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 2022, pp. 1–4. doi:10.1109/IATMSI56455.2022.10119271.
10. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM* **2014**, *61*, 95–102.
11. Khan, S.; Amin, M.B.; Azar, A.T.; Aslam, S. Towards Interoperable Blockchains: A Survey on the Role of Smart Contracts in Blockchain Interoperability. *IEEE Access* **2021**, *9*, 116672–116691. doi:10.1109/ACCESS.2021.3106384.
12. Kolb, J.; AbdelBaky, M.; Katz, R.H.; Culler, D.E. Core Concepts, Challenges, and Future Directions in Blockchain: A Centralized Tutorial; Association for Computing Machinery: New York, NY, USA, 2020; Vol. 53. doi:10.1145/3366370.