Technology Report

Blockchain: From Concepts to Application

Eero Janhonen, Fidos Oy; Sami Peura, Fidos Oy;
Joona Kauranen, University of Oulu, University of Helsinki

September 20, 2024

UNIVERSITY OF OULU
APPLIED UNIVERSITY OF OULU

Contents

1 Introduction

Blockchains enable the construction of decentralized computing environments. These environments can be conceptualized as computers. The fundamental distinction lies in decentralization, where all the computation and data storage are distributed across multiple independent nodes in a network. The open decentralized architecture ensures redundancy, resilience and security, mitigating risks associated with single points of failure.

In Chapters 2 and 3 of this report, we will examine the basics of blockchains, decentralized applications and decentralized networks. We will also discuss key factors to consider when deciding if a blockchain-based decentralized approach is right for your product. These sections serve as a foundational overview and can benefit anyone interested in blockchains including executives, project managers or marketing professionals.

After exploring the how and why of blockchains, Chapter 4 introduces a software stack that enables building decentralized applications. That section will cater specifically to developers.

Finally, in Chapter 5, we will examine the decentralized infrastructure that underpins the blockchain ecosystem. This section will be most useful for those who want to gain deeper understanding of the underlying technology or those undertaking more advanced projects that benefit from running their own infrastructure, as well as for infrastructure providers that are interested in participating in maintaining these networks as part of their business.

2 Blockchain Technology Explained

This section will provide a basic understanding of blockchains, decentralized applications, and the supporting networks.

2.1 Blockchain

Blockchains can be conceptualized as computers. The main difference between a computer and a blockchain is the fact that a blockchain consists of a decentralized network of computers [35]. The computers that create a blockchain network are connected via the internet. Users also access blockchains through the internet.

The key benefits of blockchains relate to trust. By utilizing decentralization, blockchains enable executing computer programs without relying on any single computer, person or a company for the security and correctness of a program. The enhanced security and correctness features are achieved through multiple independent computers running the same computations and comparing the results. This comparison of the results is a crucial function of blockchains and is discussed in more detail in Chapter 5.4. The security enhanced decentralized programs, commonly known as dApps, are discussed in Chapter 2.2.

Another key feature of blockchains is their openness [49]. The data stored on a blockchain is accessible to anyone, and the programs built on blockchains are available for public use. The transparency of blockchains allows anyone to verify the data and application logic. This report recognizes data availability and management in the context of blockchains as a crucial, but sometimes overlooked aspect; therefore, detailed considerations are provided in Chapters 2.3 and 4.

As the comparison to computers suggests, the use cases for blockchains are very broad. In principle, anything that can be done with traditional computers and broader computing systems can be done using blockchains [1]. The most suitable use cases are those that require a high level of security, high uptime, high level of transparency, or applications that can benefit from autonomously executing logic. Additionally, blockchains are ideal for applications that aim to enable user-owned data or those that can't be implemented using traditional services due to counterparty risks or other trust-related issues. Specific use cases discussed in Chapter 3.2 highlight how these blockchain features are being put into practice.

With this basic understanding of blockchains, the following chapters will provide a deeper look into decentralized applications and decentralized networks.

2.2 Decentralized Applications

Decentralized Applications, commonly known as dApps [8], are software applications that run on a blockchain. Unlike traditional applications which are hosted on centralized servers controlled by a single entity such as a company, dApps operate on a decentralized network of computers or nodes, where each node executes the application's code and maintains its data. Every dApp hosted on a public blockchain like Ethereum is publicly available, meaning that the program's underlying code is publicly available to be examined and verified by anyone.

Decentralized Applications consist of one or more smart contracts. Smart contracts are program code that reside and are executed on a blockchain. For simplicity's sake, it is accurate to conceptualize smart contracts as program code on blockchains. However, it is important to note that the term "smart contract" highlights some significant differences between normal program code executed on a single machine and program code executed on a decentralized blockchain. Smart contract code can be likened to a contract, in the sense that it gets executed automatically when predetermined conditions are met [35]. Practically, this means that smart contracts enable the automation of logic, that for security reasons, would normally require evaluation or other forms of manual processing.

A good illustration of the opportunities presented by the high level of automation is the adoption of blockchain technology and smart contracts in finance and banking [42]. In these sectors, blockchains are already transforming the processes of creating, tracking, and moving financial assets. This report discusses financial use cases in more detail in Chapter 3.2.1.

The subsequent chapter will delve deeper into the specifics of the decentralized networks that power decentralized applications.

2.3 Decentralized Networks

In previous sections, we have explored the benefits of decentralization, such as increased security, transparency, automation and trust. These advantages are made possible by the underlying mechanics of decentralized networks, particularly through the use of consensus mechanisms [41].

Consensus mechanisms are a key component that allows blockchain networks to operate without a central authority. They ensure that all nodes in the decentralized network agree on the validity of transactions and updates to the blockchain. This collective verification process ensures that the blockchain's integrity is maintained without relying on any single entity. By distributing responsibility across a network of participants, blockchains create an autonomous system that establishes trust through decentralized decision-making.

However, achieving full decentralization doesn't end with the blockchain networks. Many blockchain applications need to rely on external systems to interact with real-world data or handle large datasets [7]. To preserve the benefits of decentralization, it's crucial that these external systems—such as oracles and data management networks—operate as decentralized networks.

Oracle networks, for example, connect blockchains to external systems, enabling smart contracts to interact with offchain data and services. If an oracle is centralized, it introduces a single point of failure that undermines the security and trust of the entire system. Decentralized oracle networks, such as Chainlink [7], address this issue by using a network of independent nodes to reach a consensus on data provided by various providers from multiple sources. By decentralizing the consensus process, the data providing nodes, and the data sources, these networks ensure that the data remains accurate, secure, and resistant to manipulation.

In addition to oracles, decentralized data management networks can be used to handle large datasets that aren't feasible to keep on the blockchain itself. These networks provide decentralized infrastructure for managing and processing data, ensuring it remains secure, tamper-proof, and accessible. For instance, systems like Space and Time [17] combine blockchain data with offchain data and use cryptographic proofs and privacy technologies to ensure that data, computations and queries are accurate and tamper-proof. Decentralized data management networks extend the benefits of decentralization to the data layer of the blockchain ecosystem.

Using blockchains together with supporting decentralized networks, such as oracles and data management networks, ensures the benefits of decentralization can be preserved from connectivity to data management and execution.

We will examine blockchains, decentralized data management and developing decentralized applications more practically in Chapter 4.

2.4 Gas and Fees

A key distinguishing feature of decentralized networks is the costs and payments associated with decentralization. In traditional applications, service providers and end-users cover costs using traditional methods like bank payments. For example, service providers might pay for cloud-based computing, or end-users might pay for software subscriptions. For decentralized networks though, direct bank payments are often not a viable option because they would introduce a centralized single point of failure into the system. To ensure a level of security that matches the underlying security of blockchains, the costs and payments for executing actions on these networks are managed through a concept called Gas [52].

Gas or similar fees are a way for users to compensate the individual operators who maintain decentralized networks like blockchains. While writing on the blockchain always incurs a gas payment, reading data from a blockchain is free. Gas prices are determined by network-demand: during periods of high demand, gas prices increase, and during less demand, gas is cheaper. This market-driven approach to determine fees acts as a mechanism to prevent network abuse by requiring users to pay for the computational resources consumed by their transactions.

In addition to their functioning as an internal currency, gas fees create the central economic incentives that are needed for creating a secure and functional blockchain system. Individuals maintaining a blockchain system are only paid when they facilitate the transactions and smart contract executions correctly. In this manner, gas fees have an important role for users as payments, to individual node operators as compensation and on a system level as an incentive mechanism, that ensures a functional system.

While gas fees may present a barrier to entry, due to the requirement to manage a wallet and its private keys, in addition to the many steps required to purchase the gas tokens, they are an integral aspect of the decentralized economy that underpins dApps. Future improvements, such as account abstraction methods, are likely to mitigate these issues, and the future user experience of blockchains isn't expected to differ from using more traditional services like online banking.

3 Using Blockchain: Guiding Principles and Use Cases

This section provides a general understanding of how to evaluate the use of blockchain technology while also offering practical guidance for readers to determine if it suits their specific use cases. Along with outlining key principles, we'll explore real-world examples from various industries, such as finance and supply chain management, to illustrate effective applications of blockchain.

In principle, any software application can be built using blockchain technology. As we have alluded before, blockchains are comparable to computers. Blockchains like Ethereum are Turing-Complete, meaning that their blockchain-based computation is only limited by the same theoretical limitations as are all the traditional computer systems [1].

However, despite these theoretical similarities, there are meaningful differences between various computer systems, and different use cases have different requirements for their underlying solutions. The following sections aim to highlight which types of requirements are well-suited for blockchain-based decentralized systems and which are less compatible.

3.1 Benefits and Drawbacks of Decentralization

3.1.1 Benefits of Decentralization

The main benefits of blockchains stem from decentralization. Decentralization can eliminate the need for many third parties that are typically involved in business processes [38]. Broadly speaking, blockchain technology is worth considering when aiming to streamline operations. Blockchains can enhance trust-related processes or, in some cases, completely remove the need for intermediaries. Instead of relying on third parties to verify transactions, manage contracts, or maintain records, smart contracts offer a level of automation that reduces delays, minimizes human errors, and cuts costs associated with middlemen.

While new platforms—such as blockchains—have the potential to enable entirely new and previously unimaginable business models and services, blockchains can also be considered for developing decentralized alternatives to traditional centralized services. Blockchains can be especially useful for fields with increasing trust-related issues. Much like cryptographically secured messaging, blockchains have the potential to extend cryptographic security to a wide range of computing applications.

Another key benefit stemming from decentralization is the reliability of blockchain networks. For example, the most popular smart contract enabled blockchain Ethereum has a flawless uptime record, with zero downtime since its inception in 2015 [22]. In contrast to centralized systems where compromising a single service provider can compromise the whole system, in a decentralized network, an attacker would need to compromise a significant portion of the network to alter its data, making such attacks impractical.

Additionally, blockchains offer transparency related benefits, every transaction, smart contract action and variable on a blockchain is recorded on a public ledger that anyone can verify. This transparency ensures accountability and reduces the potential for fraud. Moreover, once data is recorded on the blockchain, it cannot be altered retroactively which ensures the integrity of the information.

3.1.2 Drawbacks of Decentralization

Decentralization offers numerous advantages, but it also comes with significant challenges. These may include performance and scalability issues, higher operational costs, complexity in usability and development, and problematic governance and decision-making processes. Understanding these drawbacks is crucial when considering decentralization for various applications.

Centralized systems often execute tasks faster and more efficiently than decentralized networks, which require time-consuming and computationally intensive consensus mechanisms to validate transactions. As decentralized networks grow and they experience high amounts of traffic, they can become congested, leading to slower transaction times and higher fees. For example, Ethereum has experienced significant congestion and increased gas fees during high demand multiple times during its history, mainly when high-profile projects were released on its platform with lots of onchain usage [48].

Cost is another concern. Decentralized networks are generally more expensive to run due to the need for numerous nodes to validate transactions, leading to higher maintenance costs. The energy consumption, especially in Proof of Work systems like Bitcoin, is also very high, raising environmental concerns for some [29]. For Ethereum, this has not been a concern after moving to a Proof of Stake consensus model in 2022, after which its energy consumption fell by more than 99.9 percent [37].

Complexity and usability present further challenges. DApps can be more complex for users, who must manage their own keys and wallets, because losing access to keys can result in permanent asset loss. Additionally, developing on decentralized networks requires specialized skills and knowledge, adding to development complexity and costs.

Also governance and decision-making in decentralized networks can be problematic. Unlike in centralized systems, where decisions are made quickly by a small group, decentralized networks rely on community consensus, leading to slower decision-making and difficulties in reaching agreements. This is especially challenging when rapid responses or resolutions to contentious issues are needed.

Overall, while decentralization has many benefits, its challenges must be considered for each application as not every product benefits from blockchain technology.

3.2 Use Cases

This section introduces example use cases for blockchain technology. Its objective is to provide illustrative examples of how blockchain benefits are being applied in practical scenarios.

3.2.1 DeFi

Currently the most established use case for blockchains is finance. Decentralized Finance (DeFi), a blockchain-based ecosystem of financial services, is revolutionizing how financial assets are created and managed [16].

The DeFi ecosystem offers services that are familiar from the legacy financial system; payments, borrowing, lending, earning interest and trading of assets. The benefits of DeFi include trustworthiness and efficiency. DeFi services are also guaranteed to execute as programmed, and offer settlement times measured in minutes instead of the current day to weeks [2]. In addition to the basic services, DeFi makes possible the creation of new kinds of assets through the programmability of blockchains, enabling almost limitless programming of money and assets.

Both startups and major financial institutions are already utilizing blockchains especially for asset tokenization - a process of representing real-world assets as blockchain-based tokens [36].

In addition to creation of new and betterment of existing services, DeFi enables almost barless access to financial services [15]. By removing geographical barriers and traditional gatekeepers like banks, DeFi platforms allow individuals around the world to participate in the financial system. This is particularly beneficial for those in regions with limited financial infrastructure. For instance, someone in a developing country is able to easily borrow funds for a business venture or earn interest on their savings through DeFi.

Yet another noteworthy property of DeFi is the composability of services. The open-source dApps that create the DeFi ecosystem are freely usable for any developer or company. Developers have the possibility to combine existing services or build innovative additional components on top of existing ones without any kind of permissions or agreements from the other DeFi service providers.

3.2.2 Metaverse

Blockchains allow creating open, interconnected and permanent Metaverse(s), and while they aren't best suited for running the graphics of sophisticated Metaverse worlds, they can uniquely offer the Metaverse a neutral and permanent ownership layer [30]. Blockchain-based Metaverse assets, like identities or digital twins, are directly owned and controlled by users, instead of any single entity or company. Additionally, blockchain technology allows the seamless movement of assets from a platform to another. With the Metaverse becoming an important part of both our work life and free time, blockchains can offer a secure and user-oriented alternative for the current centrally operated platforms.

3.2.3 Identity and Authentication

Decentralized identity solutions put individuals in charge of their digital identities and credentials [5]. They achieve this by processing sensitive data offchain and storing cryptographic proofs on the blockchain. This method can reduce the risks associated with centralized identity providers, ensuring greater user privacy and security. It allows for the safe storage of sensitive personal data without risks that come with intermediary parties. Moreover, it can streamline processes, such as automated identity verification for seamless and secure authentication, including KYC. A decentralized identity would also enable the creation of immutable audit trails and natively support participating in and creating decentralized autonomous organizations (DAOs) [5].

3.2.4 Supply Chain

Supply chain dApps can track the journey of products along the supply chain, ensuring transparency and authenticity, particularly in industries like food and pharmaceuticals [43]. By recording each step on the blockchain, these platforms can minimize trust between consumers, producers and sellers, and mitigate the risk of fraud or counterfeit goods.

3.2.5 Social Media

Decentralized social media platforms [11] offer censorship-resistant content sharing and monetization options, enabling users to control their data and directly monetize their creative work. By decentralizing content hosting and ownership, these platforms promote user autonomy.

4 Introducing ECS: a Software Stack for Developers

4.1 Introducing ECS

This section will provide a software stack for building decentralized applications. We will cover technologies that are used in the development of smart contracts themselves, but also technologies that enhance smart contracts with necessary capabilities like connecting to existing data and systems. In the following three chapters we will introduce the ECS software stack. The acronym ”ECS” stands for Ethereum Virtual Machine, Chainlink, Space and Time, where each of the individual components play an important role in the development of secure end-to-end decentralized blockchain-based applications.

4.1.1 EVM

The Ethereum Virtual Machine (EVM) [50] serves as the execution environment within the ECS stack, acting as the foundational platform for applications built using the stack. As discussed in the second chapter of this report, blockchains can be conceptualized as computers, and in this case, the EVM acts as the decentralized computer for ECS. The EVM can be programmed using a few different purpose-built programming languages, with Solidity being a primary example.

Originally developed as part of the Ethereum blockchain, the Ethereum Virtual Machine (EVM) is an open-source decentralized virtual machine. Its open-source nature has allowed it to become the computational environment for a variety of other blockchain systems, including Avalanche, Arbitrum, and Polygon. The other technologies of the ECS stack are similarly versatile, allowing the entire stack to be used across multiple blockchains and enabling the development of cross-chain applications.

4.1.2 Chainlink

The secure compute process of EVM-based blockchains involves a network of computers executing, comparing and coming to an agreement about the result of each computation. To ensure security and efficiency, blockchains are limited to accessing a very limited set of internal data and cannot process data outside of their own networks. Chainlink [7] is a decentralized computing platform that offers an important set of services that blockchains like Ethereum can not natively provide.

The ECS stack utilizes Chainlink for connectivity, automation and running arbitrary computations that don’t need or aren’t feasible to run on the EVM.

As mentioned, the EVM completely lacks connectivity to the outside world. Chainlink provides the necessary connectivity to any external data and systems that applications might require. For example, Chainlink can supply price data for a DeFi application, or perhaps as a more illustrative example, provide the departure time of an airplane for the purposes of a smart contract insurance that automatically compensates for delayed flights. In addition to connectivity to external systems. Chainlink CCIP, a cross-chain interoperability protocol, enables interaction between multiple blockchains, a crucial function for building cross-chain applications.

Chainlink also provides automation services for smart contracts. Since smart contracts cannot access external data, they cannot perform automated actions, such as executing some action at a specific date. In addition to time-based events, Chainlink automation enables triggering smart contract execution based on any other arbitrary condition, such as the price of an asset passing a certain threshold.

Additionally, the ECS stack uses Chainlink for offloading computations to an external network of nodes. Similarly to EVM chains and other blockchains, a consensus mechanism ensures the security of these external networks, but the ability to run computation on smaller and more flexible networks is more scalable and cost effective.

4.1.3 Space and Time

Even though data on public blockchains is public, it isn't in an easy-to-use and easily available format. Moreover, the data storage capabilities of blockchains are focused on basic blockchain transactions and storage of smart contract code, which essentially makes blockchains rather lightweight databases that are impractical to manage. To address these limitations, dApps need capabilities similar to those of traditional databases and database management systems.

Space and Time (SxT) [17] addresses these challenges by offering blockchain data in an organized easy-to-use format. Space and Time also enables dApp developers to create custom databases with arbitrary data. Finally, SxT offers tools for performing searches, analytical jobs and other functions on a combination of built-in blockchain data and custom created decentralized databases.

Importantly, SxT as a data management solution for the ECS stack provides a level of security and decentralization that doesn't undermine the benefits achieved with using a blockchain in the first place. The security model of SxT is based on decentralization and zero-knowledge technologies. Similarly to EVM and Chainlink, decentralization mitigates the trust issues related to centralized control. Zero-knowledge proofs are employed in creating proofs of the integrity of datasets and the operations performed on them, without the need to disclose critical data.

In practical terms, integrating Space and Time allows dApps to ask questions about the contents of a blockchain and custom-added data. For instance a DeFi platform might use SxT to determine how many users interacted with a platform last month, the total transaction volume, or similar metrics about a competitor.

4.1.4 Ecosystem Support and Tooling

The ECS stack can be integrated with a wide range of popular blockchains and development tools. The Ethereum Virtual Machine (EVM), which acts as the execution environment for the ECS stack, powers multiple blockchains such as Avalanche, Arbitrum, and Polygon. This compatibility means that smart contracts developed using the ECS stack can also be utilized across these other chains.

Similarly, Chainlink integrates well with EVM-compatible chains, and also enables cross-chain interactions throughout the wider blockchain ecosystem. Chainlink's ability to act as a bridging technology between blockchains and traditional systems extends the integration to virtually any system.

Space and Time (SxT), another component of the ECS stack, offers indexed data from major blockchains with support for more networks continuously added. Additionally, SxT supports the creation of custom databases and integrates with Chainlink, allowing for connection between SxT's datasets and Chainlink's external services.

The ECS stack is also compatible with leading development tools and environments. Hardhat and Foundry are popular frameworks that provide comprehensive tools for compiling, testing, and deploying EVM-based smart contracts. These environments support the development process and ensure the effective implementation of smart contracts.

The most widely used wallets, such as MetaMask, support major EVM-based blockchains, ensuring that applications built using the ECS stack are compatible with most blockchain wallets.

Additionally, OpenZeppelin's security products provide a comprehensive set of reusable Solidity components and related security tools that are useful when building with the ECS stack.

For frontend development, the ECS stack can be used with popular frontend development libraries that support EVM based chains. Most popular libraries include Ethers.js and Web3.js for Javascript and Web3.py for Python.

4.1.5 Further Learning and Getting Started

This report conceptualizes the EVM, Chainlink, and Space and Time as a comprehensive development stack. Since well-established documentation is already available for each technology, development instructions are not included here. For detailed guidance on writing Solidity smart contracts, understanding the EVM, and using Chainlink and Space and Time, or setting up a development environment, refer to the official documentation and tutorials provided by the respective platforms:

Solidity: https://docs.soliditylang.org/en/
Ethereum: https://ethereum.org/en/developers/
Chainlink: https://docs.chain.link/
Space and Time: https://docs.spaceandtime.io/docs/
Hardhat: https://hardhat.org/docs/
OpenZeppelin: https://docs.openzeppelin.com/

5 Blockchain Infrastructure

The infrastructure for blockchain apps differs quite significantly from traditional apps due to the fundamental architectural differences and requirements of each system. Before jumping into the topic, understanding the old ways is important, since hosting both still shares many similar software components.

As explained in the previous chapters, blockchain technology is built on two crucial concepts: decentralization and immutability. Data distribution across a network of nodes ensures blockchains that no single entity can control the ledger, making it tamper-proof and transparent. But to truly understand how it works, we need to go deeper into its core components and how they work together.

Before that, it's important to emphasize that less decentralized options also exist for enterprises that require more control over the network, for reasons such as confidentiality guarantees. Public blockchains rely on all aspects of decentralization and open participation because this is how they were built. Private or permissioned ledgers, sometimes also called distributed ledger platforms, are networks that are quite similar to blockchains. However, they don't offer publicly open participation, and give more control over what is executed and by whom.

At its core, blockchain is composed of nodes that validate and record transactions through a distributed ledger. Validation refers to the process in which nodes verify the integrity and accuracy of transactions before adding them to the ledger - which includes the collection of transactions, including their timestamps, receiver and sender addresses, and the amounts sent. This validation process relies on complex consensus mechanisms, such as Proof Of Work (PoW) or Proof Of Stake (PoS). Consensus mechanisms will be discussed in more detail in Chapter 5.4.

5.1 Introduction to Blockchain Infrastructure

Hosting blockchain infrastructure has many similarities to hosting any other program with high uptime requirements. A blockchain node can be viewed as a computer program, and what differentiates it from a traditional backend program, is that it's always connected to other blockchain nodes across the blockchain network. In order to run a blockchain program successfully, there has to be supporting infrastructure in place to ensure this constant, to guarantee a very high uptime. This supportive infrastructure can include SQL databases, logging and monitoring software, orchestration tools like Kubernetes, and other programs used to maintain the high availability of the blockchain program. In this section, we divide blockchain hosting into its many components, all of which play crucial but distinct roles.

5.2 Differences and Similarities

In this section we will discuss the differences and similarities between blockchain node hosting and traditional web app hosting. A blockchain node's most important requirement is high availability. To achieve this, blockchain node operators use various DevOps tools and programs that are commonly applied in other high-availability internet applications. If you are running a high-traffic, high-availability internet application in production, getting into hosting a blockchain node is not difficult from a technical hosting perspective.

One difference lies in how interacting with decentralized networks works. They require users and builders to manage their own digital assets. This is where blockchain wallets come in. They are the primary user interfaces for managing blockchain tokens, interacting with smart contracts, and creating transactions. Wallets rely on Remote Procedure Calls (RPCs) for communication with blockchain nodes, an important protocol along with its underlying infrastructure, a topic that will be discussed more later.

There are specific concepts related to blockchains that must be understood before jumping into hosting a blockchain node. Firstly, the different consensus mechanisms that are fundamental to how blockchains operate. Unlike in typical web applications, where data is usually processed and validated by one server, blockchains rely on forming a consensus from a participating network of servers to validate transactions and maintain the network's security. Depending on the type of blockchain and the consensus mechanism, the hardware, energy consumption, and security requirements can vary significantly.

Regarding the hosting infrastructure, the choice of a DevOps software stack is crucial to ensure the blockchain node remains operational and connected to the blockchain at all times. To achieve this, it is advisable to utilize auxiliary programs such as logging, monitoring, load balancing, and orchestration tools. These tools offer high availability and also make updating and managing multiple blockchain nodes much easier, an option often chosen for increasing availability even further.

After selecting and building a compatible software stack to support the blockchain node or nodes, a significant aspect of operating it involves DevOps skills and infrastructure capable of handling failures without causing substantial downtime to the node. Most blockchain node applications feature a frontend developed in popular languages like JavaScript or Python, and while a blockchain node operator might not need to build or maintain the frontend, having an understanding of how it works is advantageous for quicker troubleshooting.

More importantly, it is important to review the blockchain's own documentation. Each blockchain is unique, and each of their requirements can vary significantly. While the process of hosting a node and its supporting programs may be similar across blockchains, specifics such as the blockchain's operation, programming language, block times, and consensus mechanisms can differ, resulting in different hardware and knowledge requirements.

For example, Ethereum employs a Proof of Stake (PoS) consensus mechanism [23] and uses Solidity as its primary smart contract language. This necessitates some understanding of Solidity, the programming language of EVM-based smart contracts, to be able to interact with them if needed. In contrast, Solana uses a Proof of History (PoH) consensus mechanism and uses Rust for its smart contracts [45]. Understanding these differences is crucial for effectively hosting blockchain nodes and ensuring optimal performance, given the unique characteristics of each blockchain.

5.3 Blockchain Infrastructure and Its Roles

The blockchain infrastructure offers various roles, some more technically demanding than others. Often the term “node operator” is used, which broadly refers to individuals or entities who maintain or run blockchain nodes across various blockchain networks. What does a blockchain node operator do, and what types of node operation is there? What do I need to know to be able to operate and maintain different blockchains? In this section we will answer these questions, and introducing the most important concepts and aspects to give a deeper understanding of blockchains and operating their nodes.

A blockchain node operator helps keep blockchains running. Nodes perform different tasks, such as making calculations, solving puzzles, validating transactions, and maintaining records of transactions on blockchains. However, these processes can vary from one blockchain to another. The most widely used blockchains with highest value locked in them [13] include Ethereum, Bitcoin, Solana and private chains such as Hyperledger Fabric. There are also other important types of nodes, such as oracle nodes, which also operate outside and between blockchains. Later on, we will focus more on node operation for these more specific use cases. First, we will explore a few key concepts.

5.4 Different Blockchain Architectures

There are several established types of nodes, such as validator nodes, full nodes, light nodes, and other specialized nodes like oracle nodes. Validator nodes are used for participating in blockchain consensus mechanisms, and in creation of new blocks. Other blockchain node types like full node and light node are blockchain nodes that store blockchain data and ensure blockchain integrity. Oracle nodes and other similar specialized nodes have specialized and important tasks, but they do not participate in the consensus mechanism of blockchains.

Validator nodes can use consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS) and Proof of History (PoH). Most of the popular blockchains of today use some form of Proof of Stake or a modified Proof of Stake consensus mechanism [34].

Main job of the validator node is to verify transactions; to maintain the integrity of the blockchain. It participates in the consensus process by proposing and validating new blocks, which depends on the consensus mechanism used. Validator nodes usually earn rewards for adhering to the rules, which also requires constant operation of the node infrastructure without any major downtime.

A full node stores the entire blockchain and its transaction history. Full nodes connect to the blockchain and provide endpoints like RPCs (Remote Procedure Call) and WebSockets, which allow other nodes and users to access the blockchain. Full nodes are essential for all blockchain users and developers, as they rely on these nodes to connect to the blockchain.

Specialized nodes, such as oracle nodes, serve distinct functions beyond the core consensus and blockchain data storage roles. These nodes also require a high uptime and their own dedicated infrastructure to perform their tasks efficiently.

Grasping the differences between consensus mechanisms is crucial due to their direct influence on how blockchain networks operate. Each consensus mechanism affects node roles, transaction finality, and resource consumption differently, which influences how infrastructure is designed and maintained. The next chapter will explore these models in detail to better understand their implications for blockchain infrastructure and its management.

5.4.1 Proof of Work

Proof of Work (PoW) is the oldest consensus model, with Bitcoin, launched in 2009, being the most well-known PoW blockchain. Validators in a PoW blockchain are called miners. We will take a closer look at Bitcoin's PoW mechanism, as it gives an excellent in-depth demonstration how a decentralized ledger can operate.

Miners in the Bitcoin network validate and add new blocks by solving cryptographic puzzles. Miners generate a huge number of hashes to find one that meets the required block difficulty, which in turn results in a block reward if guessed first. Each hash is calculated from the block header, which includes data such as the previous block hash, a timestamp, and a nonce - a number that miners increment with each guess to find the correct hash. The block header, including the nonce, is passed through the SHA-256 cryptographic hash function twice to produce a 256-bit hash. The goal is to produce a hash output that is lower than the predetermined target value, which is adjusted by the Bitcoin network approximately every two weeks. If a miner's resulting hash is below the target difficulty, the block is considered valid, and the first miner to discover it is rewarded with newly minted bitcoins and the transaction fees from that block [40].

Miners use specialized software to perform these operations. Examples include CGMiner, BFGMiner, and Multiminer [39]. This software interfaces with the Bitcoin network to submit newly found blocks and manage mining tasks. It also communicates with other nodes to propagate the block across the network. Once a block is broadcasted, other nodes verify its validity by checking the proof of work and the transactions within it. Consensus is achieved when the majority of nodes agree on the validity of the new block. These steps ensure that reaching consensus always requires widespread agreement.

5.4.2 Proof of Stake

Ethereum is the most known Proof of Stake blockchain, although this wasn't the case when it launched in 2015 using Proof of Work. In The Merge in 2022, Ethereum successfully transitioned from PoW to PoS, increasing scaling and decreasing the network's energy consumption [26]. While there are many other PoS chains, we will focus on Ethereum as it has the longest and most tested history of producing blocks [22].

In PoS blockchains, a validator gets rewarded for validating transactions. Ethereum blockchain consists of two main parts: the execution layer and the consensus layer. The execution layer is responsible for executing smart contracts, processing transactions, and maintaining the state of the Ethereum network. Meanwhile, the consensus layer, called Beacon Chain, handles the network's security and agreement on the state of the blockchain. More about the consensus layer and Beacon Chain in a later section.

In Ethereum's PoS system, validators are required to first stake ETH as collateral for the Ethereum deposit smart contract; they then are randomly selected, proportional to their stake, to propose and validate new blocks. When a validator is selected to propose a block, a block containing recent transactions (from the execution layer) is submitted to the network. Other validators review the proposed block and vote on its validity, a process that is called attestation. Once enough attestations have been received and a finality checkpoint is reached, the block is added to the blockchain, at which point it is considered permanent and secure. Validators are rewarded for honest behavior and penalized, "slashed" for malicious or negligent behavior. We will discuss the topic of blockchain validation economics in more detail in Chapter 5.5.

For validators, staked ETH is subject to a lock-up period. Once ETH is deposited into the staking contract, it is locked up and cannot be withdrawn until certain conditions are met [24]. This lock-up period helps ensure that validators have a long-term commitment to the network. Validators can withdraw their staked ETH only after meeting withdrawal eligibility criteria, which is designed to prevent sudden or large-scale withdrawals that could disrupt network stability. The lock-up mechanism also includes a waiting period after validators request withdrawal before their ETH is actually released to enhance network stability.

The consensus mechanism is managed by the Beacon Chain, used by validators to coordinate the PoS protocol and to manage validator operations [25]. The Beacon Chain oversees staking, slashing, and finality processes for Ethereum. The execution mechanism, managed by Ethereum clients such as Go-Ethereum, is used to interact with the Beacon Chain and the Ethereum blockchain. These clients handle tasks such as submitting transactions, querying blockchain data, and participating in consensus. While many validators run their own full Ethereum nodes to maintain decentralization, some may rely on third parties for this for convenience. However, relying on third parties in this way can undermine decentralization by concentrating data access and validation functions through fewer external services, creating potential single points of failure.

5.4.3 Other Consensus Mechanisms

Solana, a blockchain launched in 2020, uses Proof of History (PoH) alongside Proof of Stake to validate and add new blocks. Unlike traditional consensus mechanisms, PoH provides a historical record that proves an event has occurred at a specific moment in time. This is achieved by generating a verifiable sequence of hashes, where each hash depends on the previous one. This sequence acts as a historical record that shows the exact timing and order of events. PoH is used by Solana to streamline transaction processing by integrating timestamping directly into the protocol [51].

In essence, Proof of History introduces a cryptographic timestamp to the blockchain. This timestamp allows the network to efficiently order transactions and to provide a verifiable historical record of events, which streamlines the validation process, allowing a theoretical throughput of 710k transactions-per-second [51].

Validators in Solana are selected based on the amount of SOL, the token of Solana, staked as collateral in a staking smart contract, which is required for participation. This is done by locking up SOL in a staking contract which supports the network's security and operations. The tokens have a lock up period of one "epoch", a certain amount of blocks, during which time they can't be accessed [46]. Validators are chosen to propose new blocks and confirm the validity of transactions based on the size of their stake, with those staking more SOL having a higher probability of being selected.

When a validator node is selected as a “leader node”, they propose a block that includes transactions and the cryptographic proofs of historical events provided by the PoH mechanism. Other validators then review the proposed block and vote on its validity. This voting process is designed to be efficient and leverages the PoH record to quickly assess the order and consistency of transactions. Once a sufficient number of validators have agreed on the validity of a block, it is added to the blockchain.

Non-validator nodes in Solana are called RPC nodes. They relay information, provide data access, and support network operations without participating in consensus. Running an RPC node allows interaction with the Solana network without the need to stake SOL or participate in validation. This can be useful for accessing network data [44].

5.4.4 Oracle Nodes

As mentioned in Chapter 2.3, Chainlink is a decentralized oracle that provides smart contracts with external data with the help of decentralized oracles. Unlike blockchains such Ethereum or Bitcoin, Chainlink is not a standalone blockchain, but a middleware network that operates between blockchains and the real world, by delivering offchain data to blockchains.

Chainlink's core components are the Chainlink nodes, which are operated by independent entities. These nodes fetch and deliver data from off-chain sources to the blockchain. Node operators are rewarded with LINK tokens for providing accurate data, and the nodes will face penalties for errors or malicious behavior in later software releases.

Chainlink utilizes smart contracts to handle data requests and aggregation. When a smart contract requires data, multiple oracles provide independent responses. The Chainlink protocol aggregates these responses to produce a final result, ensuring accuracy and reliability.

As per Chainlink's whitepaper [7], the reputation system tracks the performance of node operators, rewarding those with a history of reliable data delivery. LINK tokens are used to pay for data requests and incentivize nodes. They can also be staked as collateral to ensure node operators fulfill their obligations. Chainlink nodes require a high availability setup to avoid downtime [10].

5.4.5 Non-validator Nodes

Non-validator nodes are crucial components of a blockchain network. Unlike validator nodes, they do not validate new transactions and do not receive rewards. Instead, they verify the history of the transactions by maintaining a number of the most recent blocks of the blockchain. They also store the state of the blockchain.

Running a non-validator node exposes connection endpoints, to which users and other nodes can connect to access the data of the blockchain. There are two main endpoint types - RPCs (Remote Procedure Calls) and WebSockets. RPCs allow remote communication between programs or computers, and they allow node operators and developers to query, submit transactions, and interact with smart contracts without running a personal node [3]. RPC endpoints are categorized into two types: public endpoints, which are free but less secure and rate-limited, and private endpoints, which are paid and offer enhanced performance, security, and reliability.

WebSocket, on the other hand, provides a two-way communication channel over a single TCP connection. It is favored in blockchain applications because it allows for real-time updates, immediately reflecting changes from the blockchain in the connected interface.

In summary, while non-validator nodes don't earn rewards in the same way validators do, they are essential for maintaining blockchain integrity and providing access through connection protocols such as the RPC and WebSocket.

5.4.6 Private Blockchains

The term "private blockchain" is used here for clarity, though "private chain" or "private ledger" is more commonly used. Despite the terminology, several private blockchain solutions exist, with Hyperledger Fabric being one of the most popular. Fabric offers customizable platforms for enterprises, allowing companies to tailor nearly every aspect of the blockchain to their specific needs [31].

Running a node on a private blockchain differs from public networks since participants in private blockchains are typically trusted entities. This trust allows for more flexible governance and operational models, but it also means that node operation varies across different blockchains, depending on how the network is structured. Therefore there is no specific way how to run blockchain node in private blockchains because all specifications can vary depending how all blockchain entities want that blockchain to operate. Enterprises use private chains to get control, customization and security to their own blockchain solutions. For example, IBM Food Trust uses a private, permissioned blockchain built on IBM Blockchain built on top of Hyperledger Fabric. This platform enhances food safety and traceability by allowing only authorized participants to access and share data securely across the supply chain [32].

5.4.7 Technical Requirements

There are different technical requirements for different blockchain nodes. This results in varying amounts of operation costs for operators of different nodes. Bitcoin validator nodes, also called miner nodes, are usually run by specialized computers made for computing hashes, called ASIC miners. These miners generate immense hash rates at the cost of high power usage. Examples of such hardware include the Bitmain Antminer and MicroBT WhatsMiner models [14]. For Proof of Stake validator nodes like Ethereum, both the client software components - consensus and execution layer, come with recommendation to have a fast CPU with 4+ cores, 32 GB RAM, 2 TB SSD and a 1Gbps internet speed [4]. An in-depth explanation of running an Ethereum node will be given in a later section.

For other blockchain node operations, such as running a Chainlink node, recommended hardware includes a 4-core CPU, at least 16 GB of RAM, and a reliable internet connection. This setup supports data processing and ensures the node operates efficiently within the network [9]. PostgreSQL, the database used by the Chainlink nodes for storing node data, requires an additional system with a 4 core CPU, 16 GB RAM and 100 GB of storage. There are many other types of blockchain nodes and their specification varies a lot.

5.5 Economics of Blockchain Hosting

For a network to be called a blockchain, there must be multiple independent nodes hosting the distributed ledger and processing transactions. Most blockchains need a significant number of nodes to ensure reliability and viability for users. While the development company or organization behind a blockchain could host all the necessary nodes themselves, this would seriously undermine the key advantage of decentralization. Therefore, blockchains often recruit third-party node operators to validate transactions and host the blockchain's data. However, third-party companies, individuals, and other entities are generally not willing to host blockchain nodes without any financial compensation. This is why blockchains offer various incentives, typically in the form of profit expectations for node operators as tokens minted by the blockchain or protocol.

Developers, on the other hand, might lack the interest or expertise to operate their own blockchain nodes, which is when they will pay for access to existing nodes via RPC connections. Node operators can monetize this by charging fees to developers for this RPC connection access, creating a revenue stream. In this way, developers pay for the infrastructure they don't want to host themselves, and node operators can generate income by providing reliable access to blockchain networks.

This next section will cover the financial risks, costs, and ways to generate revenue from blockchain infrastructure hosting. Monetizing node operations can be complex and often involves models uncommon in traditional finance. Common methods include reward models, direct payouts, and nodes-as-a-service. Each has its own financial structure, allowing node operators to build business models around them.

5.5.1 Reward Models

Reward models include mechanisms such as validator rewards, staking rewards, and delegation rewards. These models compensate node operators based on specific rules of the blockchain, such as blockchain usage, staking amounts, and node reliability (uptime). Typically, validators are rewarded with the native token or cryptocurrency of the blockchain, such as ETH in Ethereum. These tokens are often converted into fiat currency to cover node upkeep costs and generate profit. Reward models are commonly used in public blockchains and are effective primarily on production chains, or 'mainnets,' with a publicly tradable token.

Blockchains aim to provide financial incentives to their validators. Many blockchains use Proof of Stake models, where participation is open to anyone who has staked a sufficient amount of the blockchain's token or cryptocurrency. Staking is the most commonly used reward model and provides rewards based on the amount of tokens each node operator has staked in the staking pool. Most chains require blockchain nodes to stake tokens to receive rewards for validating transactions. These nodes are known as validator nodes. For example, on Ethereum, a validator node operator must stake at least 32 ETH to participate in blockchain's consensus mechanism [18]. Blockchain validator rewards typically range between 3 and 8 percent APR per year [28]. While this might not seem like a lot for validators who stake the minimum requirement, profitability increases as the staked amount grows because there is no need to host additional nodes when increasing the stake.

One way to improve the financial viability of validator is through delegation. Delegation allows entities, individuals, or companies to stake their tokens with a validator node. This enables the validator node operator to earn rewards without needing to own a large amount of the protocol's tokens. In this model, node operators earn a fee from the rewards collected. For example, if the chain offers a 5 percent APR, the fee for the validator can vary from 0.1 to 1 percent, with the delegator receiving the majority of the rewards and the validator keeping the remainder. Delegation is not financially viable unless the delegation pool is substantial, allowing the validator to earn significant fees. Delegated nodes often manage large amounts of the protocol's cryptocurrency, sometimes worth hundreds of millions of dollars [6]. This setup allows validators to receive substantial rewards without having a large amount of capital locked as collateral. However, a large delegated portion does increase risks, such as slashing, which will be discussed in later sections.

5.5.2 Node as a Service

Node as a Service (NaaS) is a service model where a node operator sells access to their node, essentially offering it as a service. This model is similar to Software as a Service (SaaS), where customers (in this case, users of the node) pay a subscription fee to access the node. NaaS operators typically offer RPC connections to blockchain nodes. Examples include Infura and Fiews.io, which host blockchain nodes and provide RPC and Websocket connections to users who build and interact with blockchain applications [27, 33].

NaaS is an effective way to monetize blockchain nodes in test networks, which generally do not provide monetary incentives to node operators unless the operator is directly compensated by the blockchain's development team.

5.5.3 Direct Payout

When the blockchain development entity pays directly to the node operator for hosting a node in their blockchain, this is a direct payout. Payments can be made in the native token of the protocol or in fiat currency. Direct payouts are more common when operating on blockchains that do not have a publicly tradable cryptocurrency, such as with private or test chains, where payments are often made in fiat currency.

When payments are made in tokens, the protocol's development company pays node operators and validators directly in tokens. The distribution of the tokens can be publicly disclosed, and there is typically a business-to-business (B2B) relationship between the payer and the node operator. In these cases, accepting payments in tokens can be riskier for validators, as many protocols devalue their tokens quickly compared to fiat currencies like the US dollar, potentially causing financial losses for node operators. However, if the protocol succeeds significantly, the tokens may appreciate greatly in fiat value, allowing validators to benefit from the increase.

These models are often used in new projects to provide sufficient financial incentives for validators and to build trust with validators in the early stages of the protocol, as there are usually only a few blockchain nodes or validators at this stage. From the blockchain development company's perspective, direct payments are useful when the blockchain does not yet have a native token but still requires nodes to validate transactions and host data until the network matures. Without direct payments, node operators would have to run the nodes without revenue, incurring a loss, making direct payments necessary.

5.5.4 Cost of Infrastructure Hosting

Running a blockchain node involves multiple costs, including server expenses, labor hours, blockchain gas fees, token purchases, and the volatility of token prices. The costs associated with blockchain node infrastructure depend on the specific programs and additional resources required to ensure high availability. For the blockchain node program itself, hardware requirements can range significantly. As mentioned before, blockchain nodes typically need computers with 4 to 16 cores, 16 to 256 GB of RAM, and up to terabytes of storage per node [47, 4]

Blockchain-specific costs include token requirements for validators or delegation and blockchain fees related to validation activities. For example, Ethereum requires a minimum of 32 ETH staked for a validator, which translates to a substantial upfront cost, amounting to almost a hundred thousand euros as of the time of writing.

5.5.5 Blockchain Fees

Fees vary depending on the blockchain and its level of activity. For instance, in Ethereum, transaction fees increase when blockchain activity is high [52]. In some cases, node operators are responsible for covering these fees themselves from performing tasks on the blockchain. Operational costs can fluctuate based on activity levels, which is particularly relevant for blockchain oracle node operators like Chainlink. In private ledgers or test network blockchains, fees are generally not applicable in fiat terms, as tokens are often provided free of charge.

Other potential costs, such as slashing and token price fluctuations, will be explored in the next section.

5.5.6 Monetary Risks and Challenges

Blockchains operate in a rapidly evolving technological sector, which introduces new untested technologies, security vulnerabilities, and other uncharted challenges. Concepts such as slashing and token price movements introduce additional complexities that require careful planning and management. These factors affect almost all blockchain node operators, especially those involved in validator node operation and participating in reward models.

Slashing is a feature commonly found in Proof of Stake protocols and similar consensus mechanisms. It refers to penalties imposed on nodes that fail to participate correctly in protocol tasks, such as new block validation or network pings. The consequences of slashing can include losing some of the node's accumulated rewards, losing validator status, downtime, or forfeiting a portion of staked tokens. In practice, slashing often results in validators (blockchain node operators) losing some of their staked tokens and/or accumulated rewards [21]. This penalty is intended to ensure that nodes perform their duties correctly and maintain high uptime. If a node fails to validate transactions or blocks properly, or if it submits “bad blocks" (blocks containing malicious or erroneous transactions), it faces slashing. Bad blocks are usually the result of either intentional manipulation or misconfiguration.

Token price action presents another challenge for blockchain node operators. Validators must hold a stake in the blockchain’s native token, and all rewards are also issued in this token. Consequently, the fiat value of rewards can fluctuate between the time they are earned and when they are converted into fiat currency. In many blockchains, the minimum stake required for validation is fixed in terms of the blockchain’s token, so the cost of staking depends on the token’s price. Node operators might attempt to “time the market” by purchasing tokens when prices are low, but this is risky as predicting market movements is inherently uncertain. Large validation stakes can lead to significant value fluctuations, potentially resulting in financial losses.

To mitigate the risk associated with token price fluctuations, one strategy is to convert all received rewards into fiat currency immediately. While this approach reduces potential profit margins if the token appreciates, it eliminates downside risk. Alternatively, node operators might choose to liquidate enough tokens to cover hosting costs and convert 75 percent of the profits into fiat, while retaining 25 percent in the blockchain’s native token. The balance between these strategies depends on the operator’s risk tolerance.

5.6 Ethereum Node Hosting

In this section, we will provide an example scenario for hosting an Ethereum node, including what is needed to successfully set up and maintain it. Ethereum is one of the most widely used blockchains, both in terms of value locked and as a smart contract platform. It is the second-largest blockchain by the total valuation of its native token, ETH [13]. Ethereum also serves as the foundation for the Ethereum Virtual Machine (EVM), the most commonly used smart contract execution environment, which many other blockchains like Polygon, Arbitrum and Avalanche also leverage [12].

Ethereum has two primary components: the execution layer and the consensus layer. When referring to an "Ethereum node," it typically means both layers working together, as discussed previously.

Ethereum nodes come in three main types: full nodes, light nodes, and archive nodes [20]. Full nodes validate the blockchain block-by-block, downloading and verifying both block body and state data. They can start verification from the genesis block or a more recent trusted block, but they only store recent data locally, typically the last 128 blocks. This allows them to save disk space by deleting older data, which can be regenerated if needed. Full nodes participate in block validation, verify all blocks and states, and serve the network by providing data when requested.

Archive nodes are a type of full nodes that never deletes any downloaded data, storing everything from the genesis block and building an archive of historical states. This makes them useful for querying historical data, such as account balances at specific blocks, and for services like block explorers and chain analytics.

Light nodes, in contrast, store only a subset of the blockchain data necessary for transaction verification, relying on full nodes for information. This makes them less resource-intensive and quicker to sync with the network, making them ideal for users who want to interact with Ethereum without having to download the entire blockchain.

Choosing to run a validator node, which is used for staking Ethereum and validating transactions, allows you to participate in the reward system. After choosing the node type, you must select a consensus layer client. Options include Lighthouse, which is Rust-based and known for its performance orientation but is more complex to use; Prysm, which is Go-based and renowned for its ease of use; Teku, which is preferred by institutional users; and Nimbus, a lightweight client suitable for resource-constrained devices.

For the execution layer client, you have options like Geth, a Go-based and the most popular execution layer client; Nethermind, known for its performance; or Besu, which targets enterprise users [19].

In this example, we use Geth as the execution layer client and Lighthouse as the consensus layer client. Syncing both layers involves aligning them to process transactions simultaneously. This syncing process can take up to several hours for a Light Node and even days for a Full Node. Once synced, the blockchain node is operational. If you are running a validator node, you will need to transfer Ethereum into the node's staking wallet to start staking. Detailed instructions for setting up Geth and Lighthouse can be found on their respective websites:

Lighthouse: https://lighthouse-book.sigmaprime.io/
Geth: https://geth.ethereum.org/docs
Ethereum Guide: https://ethereum.org/en/run-a-node/

To ensure high availability for your Ethereum RPC/Websocket endpoint, you should run multiple full or light Ethereum Nodes and use a reverse proxy to consolidate access to a single port. For validator nodes, implementing an automated startup solution is advisable to handle failures and ensure that only one validator node is active at any time, avoiding slashing due to double signatures.

6 Ending Notes

In this report, we have covered several important aspects of the blockchain space. The primary aim is to provide tools for innovation across various industries. For developers, it serves as a starting point for building new products, while infrastructure providers can utilize the report to expand into new domains.

Additionally, the themes presented here reflect a broader, forward-looking approach. By understanding the underlying principles, technological architecture, and practical applications of blockchain, readers can begin to grasp its full potential. As decentralized systems are increasingly integrated into mainstream infrastructure, they will fundamentally alter how industries function, distributing control and decision-making across decentralized networks. This shift generates fresh opportunities for innovation, enabling individuals and organizations to explore new possibilities.

Bibliography

[1] O. R. Adrian. “The blockchain, today and tomorrow”. In: *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. 2018, pp. 458–462.

[2] N. Agarwal, P. Wongthongtham, N. Khairwal, and K. Coutinho. “Blockchain application to financial market clearing and settlement systems”. In: *Journal of Risk and Financial Management* 16.10 (2023), p. 452.

[3] Ankr. *RPCs and APIs: Key Differences and Their Role in Web3 Development*. Accessed: 2024-09-20. 2024. URL: https://www.ankr.com/blog/rpc-vs-api/.

[4] BaCloud. *Ethereum Node Server Requirements 2024*. Accessed: 2024-09-20. 2024. URL: https://www.bacloud.com/en/knowledgebase/203/ethereum-node-server-requirements-2024.html.

[5] P. Bai and C. Bisht. “Decentralized Identity Management: Prerequisiteof Web3 Identity Model”. In: *Authorea Preprints* (2023).

[6] S. Beach. *Validators*. Accessed: 2024-09-20. 2024. URL: https://solanabeach.io/validators.

[7] L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magauran, D. Moroz, et al. “Chainlink 2.0: Next steps in the evolution of decentralized oracle networks”. In: *Chainlink Labs* 1 (2021), pp. 1–136.

[8] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung. “Decentralized applications: The blockchain-empowered software system”. In: *IEEE access* 6 (2018), pp. 53019–53033.

[9] Chainlink. *Chainlink Node Requirements*. Accessed: 2024-09-20. 2024. URL: https://docs.chain.link/chainlink-nodes/resources/requirements.

[10] Chainlink. *Security and Operation Best Practices*. Accessed: 2024-09-20. 2024. URL: https://docs.chain.link/chainlink-nodes/resources/best-security-practices.

[11] Chainstack. *Decentralized Social Networks*. Accessed: 2024-09-20. 2024. URL: https://chainstack.com/decentralized-social-networks/.

[12] Coinbase. *What is the Ethereum Virtual Machine (EVM)?* Accessed: 2024-09-20. 2024. URL: https://www.coinbase.com/learn/crypto-glossary/what-is-the-ethereum-virtual-machine.

[13] CoinGecko. *Top Blockchains Ranked by Total Value Locked (TVL)*. Accessed: 2024-09-20. 2024. URL: https://www.coingecko.com/en/chains.

[14] CoinLedger. *Best Bitcoin Mining Hardware*. Accessed: 2024-09-20. 2024. URL: https://coinledger.io/tools/best-bitcoin-mining-hardware.

[15] L. W. Cong, K. Tang, Y. Wang, and X. Zhao. *Inclusion and democratization through web3 and defi? initial evidence from the ethereum ecosystem.* Tech. rep. National Bureau of Economic Research Cambridge, MA, USA, 2023.

[16] S. Dos Santos, J. Singh, R. K. Thulasiram, S. Kamali, L. Sirico, and L. Loud. "A new era of blockchain-powered decentralized finance (DeFi)-a review". In: *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE. 2022, pp. 1286–1292.

[17] S. Dykstra, J. White, N. Holiday, C. Daly, D. Alves, and I. Joiner. *Space and Time: The Verifiable Compute Layer for Web3*. Accessed: 2024-09-20. Jan. 2024. URL: https://assets-global.website-files.com/642d91209f1e772d3740afa0/658edf3cf26933c4878ec965_whitepaper.pdf.

[18] Ethereum. *Ethereum Staking*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/staking/.

[19] Ethereum. *Full Node*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/developers/docs/nodes-and-clients/#full-node.

[20] Ethereum. *Nodes and Clients*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/developers/docs/nodes-and-clients/.

[21] Ethereum. *Proof-of-Stake Rewards and Penalties*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/.

[22] *Ethereum Blocks - Etherscan*. https://etherscan.io/blocks. Accessed: 2024-09-20.

[23] Ethereum Foundation. *Proof of Stake (PoS) Consensus Mechanism*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.

[24] Ethereum Foundation. *Staking Withdrawals*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/staking/withdrawals/.

[25] Ethereum Foundation. *The Beacon Chain*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/roadmap/beacon-chain/.

[26] Ethereum Foundation. *The Merge*. Accessed: 2024-09-20. 2024. URL: https://ethereum.org/en/roadmap/merge/.

[27] Fiews. *Fiews: Blockchain Connectivity as a Service*. Accessed: 2024-09-20. 2024. URL: https://fiews.io/.

[28] HackerNoon. *Step-by-Step Guide: How to Be a Blockchain Validator and Earn Rewards from It*. Accessed: 2024-09-20. 2024. URL: https://hackernoon.com/step-by-step-guide-how-to-be-a-blockchain-validator-and-earn-rewards-from-it.

[29] A. N. Q. Huynh, D. Duong, T. Burggraf, H. T. T. Luong, and N. H. Bui. "Energy consumption and Bitcoin market". In: *Asia-Pacific Financial Markets* 29.1 (2022), pp. 79–93.

[30] T. Huynh-The, T. R. Gadekallu, W. Wang, G. Yenduri, P. Ranaweera, Q.-V. Pham, D. B. da Costa, and M. Liyanage. “Blockchain for the metaverse: A Review”. In: *Future Generation Computer Systems* 143 (2023), pp. 401–419.

[31] Hyperledger. *Hyperledger Fabric Model.* Accessed: 2024-09-20. 2024. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.5/fabric_model.html.

[32] IBM. *IBM Supply Chain Intelligence Suite: Food Trust.* Accessed: 2024-09-20. 2024. URL: https://www.ibm.com/products/supply-chain-intelligence-suite/food-trust.

[33] Infura. *Infura: Web3 Development Platform.* Accessed: 2024-09-20. 2024. URL: https://www.infura.io/.

[34] Investopedia. *What Are Consensus Mechanisms in Blockchain and Cryptocurrency?* Accessed: 2024-09-20. 2024. URL: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp.

[35] L. Ismail and H. Materwala. “A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions”. In: *Symmetry* 11.10 (2019), p. 1198.

[36] A. A. Juan, E. Perez-Bernabeu, Y. Li, X. A. Martin, M. Ammouriova, and B. B. Barrios. “Tokenized Markets Using Blockchain Technology: Exploring Recent Developments and Opportunities”. In: *Information* 14.6 (2023), p. 347.

[37] E. Kapengut and B. Mizrach. “An event study of the ethereum transition to proof-of-stake”. In: *Commodities* 2.2 (2023), pp. 96–110.

[38] M. Klems, J. Eberhardt, S. Tai, S. Härtlein, S. Buchholz, and A. Tidjani. “Trustless intermediation in blockchain-based decentralized service marketplaces”. In: *Service-Oriented Computing: 15th International Conference, ICSOC 2017, Malaga, Spain, November 13–16, 2017, Proceedings*. Springer. 2017, pp. 731–739.

[39] Koinly. *5 Best Crypto Mining Software Programs September 2024*. Accessed: 2024-09-20. 2024. URL: https://koinly.io/fi/blog/best-crypto-mining-software/.

[40] S. Nakamoto and A. Bitcoin. “A peer-to-peer electronic cash system”. In: *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf* 4.2 (2008), p. 15.

[41] D. P. Oyinloye, J. S. Teh, N. Jamil, and M. Alawida. “Blockchain consensus: An overview of alternative protocols”. In: *Symmetry* 13.8 (2021), p. 1363.

[42] P. K. Ozili. “Decentralized finance research and developments around the world”. In: *Journal of Banking and Financial Technology* 6.2 (2022), pp. 117–133.

[43] A. Rejeb, K. Rejeb, S. Simske, and J. G. Keogh. “Exploring blockchain research in supply chain management: A latent Dirichlet allocation-driven systematic review”. In: *Information* 14.10 (2023), p. 557.

[44] Solana. *EVM to SVM: Client Differences.* Accessed: 2024-09-20. 2024. URL: https://solana.com/developers/evm-to-svm/client-differences.

[45] Solana. *Overview of Developing On-chain Programs.* Accessed: 2024-09-20. 2024. URL: https://solana.com/docs/programs/overview.

[46] Solana. *Staking: Delegation Timing Considerations*. Accessed: 2024-09-20. 2024. URL: https://solana.com/staking#delegation-timing-considerations.

[47] Solana Labs. *Solana Validator Requirements*. Accessed: 2024-09-20. 2024. URL: https://docs.solanalabs.com/operations/requirements.

[48] M. Spain, S. Foley, and V. Gramoli. "The impact of ethereum throughput and fees on transaction latency during icos". In: *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2020.

[49] J. Sunny, N. Undralla, and V. M. Pillai. "Supply chain transparency through blockchain-based traceability: An overview with demonstration". In: *Computers & Industrial Engineering* 150 (2020), p. 106895.

[50] G. Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[51] A. Yakovenko. "Solana: A new architecture for a high performance blockchain v0. 8.13". In: *Whitepaper* (2018).

[52] A. A. Zarir, G. A. Oliva, Z. M. Jiang, and A. E. Hassan. "Developing cost-effective blockchain-powered applications: A case study of the gas usage of smart contract transactions in the ethereum blockchain platform". In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30.3 (2021), pp. 1–38.