2
Decentralization

Decentralization is not a new concept. It has been used in strategy, management, and the government sector for a long time. The basic idea of decentralization is to distribute control and authority to the peripheries of an organization instead of it being concentrated in one central body. This structure produces several benefits for organizations, such as increased efficiency, expedited decision making, better motivation, and a reduced burden on upper management.

In this chapter, we will discuss the concept of decentralization in the context of blockchain, the fundamental basis of which is that no single central authority controls the network. This chapter will present examples of various methods of decentralization and ways to achieve it. Furthermore, we will discuss decentralized applications and platforms for achieving decentralization.

In this chapter, we will cover the following:

- Introducing decentralization
- Full ecosystem decentralization
- Decentralization in practice
- Innovative trends

Introducing decentralization

Decentralization is a core benefit of blockchain technology. By design, blockchain is a perfect vehicle for providing a platform that does not need any intermediaries and that can function with leaders chosen over time via consensus mechanisms. This model allows anyone to compete to become the decision-making authority. This competition is governed by a consensus mechanism, which we will discuss in *Chapter 5, Consensus Algorithms*.

Decentralization is applied in varying degrees from a semi-decentralized model to a fully decentralized one depending on the requirements and circumstances. Decentralization can be viewed from a blockchain perspective as a mechanism that provides a way to remodel existing applications and paradigms, or to build new applications, giving full control to users.

IT infrastructure has conventionally been based on a centralized paradigm whereby database or application servers are under the control of a central authority, such as a system administrator. With Bitcoin and the advent of blockchain technology, this model has changed, and now the technology exists to allow anyone to start a decentralized system and operate it with no single point of failure or single trusted authority. It can either be run autonomously or by requiring some human intervention, depending on the type and model of governance used in the decentralized application running on the blockchain.

The following diagram shows the different types of systems that currently exist: central, distributed, and decentralized. This concept was first published by Paul Baran in *On Distributed Communications: Introduction to Distributed Communications Networks* (Rand Corporation, 1964):

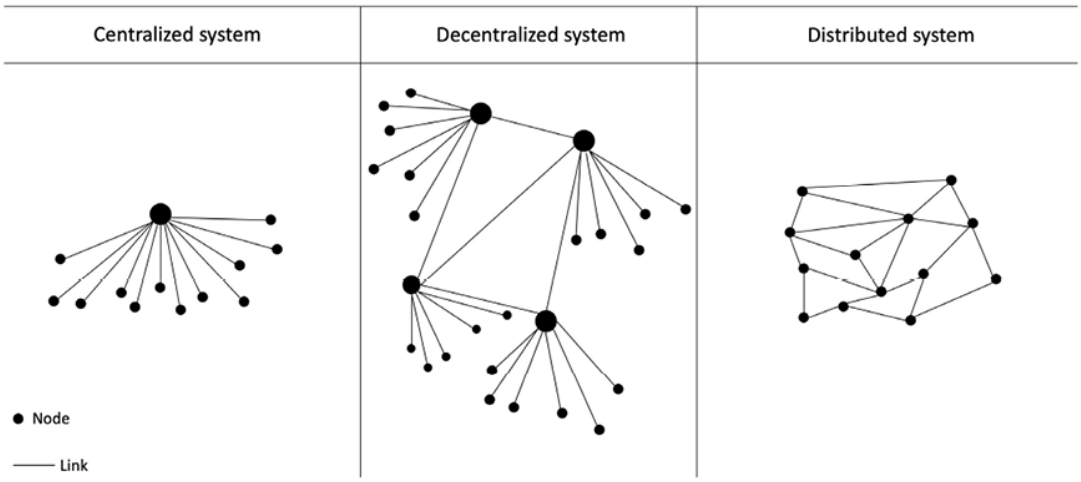

Figure 2.1: Different types of networks/systems

Centralized systems are conventional (client-server) IT systems in which there is a single authority that controls and is solely in charge of all operations on the system. All users of a centralized system are dependent on a single source of service. Most online service providers, including Google,

Amazon, eBay, Yahoo!, and Apple's App Store, use this conventional model to deliver services.

In a **distributed system**, data and computation are replicated across multiple nodes in a network in what users view as a single, coherent system.

> Sometimes, this term is confused with *parallel computing*. Variations of both models are used to achieve fault tolerance and speed. While there is some overlap in the definition, the main difference between these systems is that in a parallel computing system, computation is performed by all nodes simultaneously to achieve a single result; for example, parallel computing platforms are used in weather research and forecasting, simulation, and financial modeling. In the parallel system model, there is still a central authority that has control over all nodes and governs processing. This means that the system is still centralized in nature.

A **decentralized system** is a type of network where nodes are not dependent on a single master node; instead, control is distributed between many nodes. This is analogous to a model where each department in an organization is in charge of its own database server, thus taking away the power from the central server and distributing it to the sub-departments, who manage their own databases.

A significant innovation in the decentralized paradigm is **decentralized consensus**. This mechanism came into play with Bitcoin, and it enables a user to agree on something via a consensus algorithm without the need for a central, trusted third party, intermediary, or service provider.

We can also now view the different types of networks from a different perspective, where we highlight the controlling authority of these networks as a symbolic hand, as shown in the following diagram:

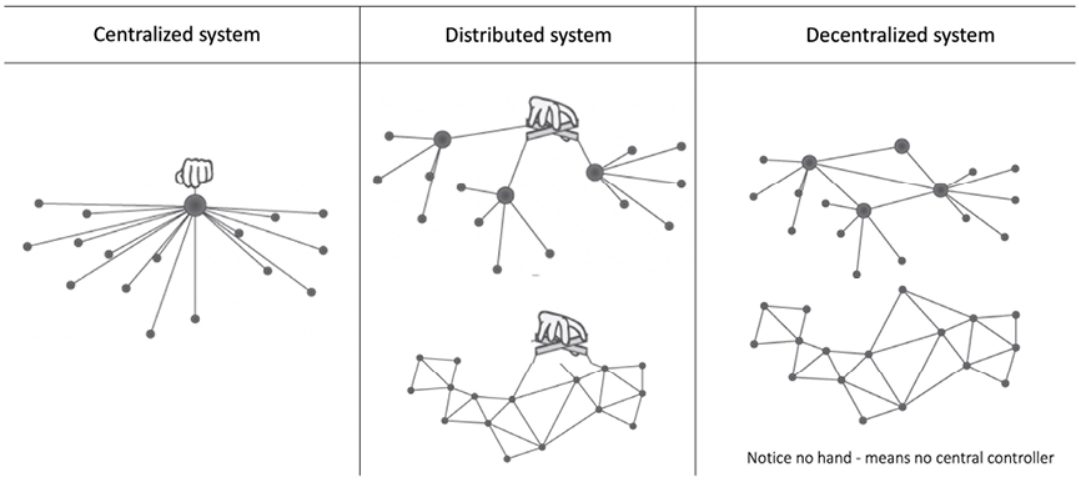

Figure 2.2: Different types of networks/systems depicting decentralization from a modern perspective

The preceding diagram shows the traditional centralized model with a central controller, which represents the usual client/server model. In the middle, we have distributed systems, where we still have a central controller, but the system comprises many dispersed nodes. On the right-hand side, notice that there is no hand/controller controlling the networks. This is the key difference between decentralized and distributed networks. A decentralized system may look like a distributed system from a topological point of view, but it doesn't have a central authority that controls the network.

The differences between distributed and decentralized systems can also be viewed at a practical level in the following diagram:

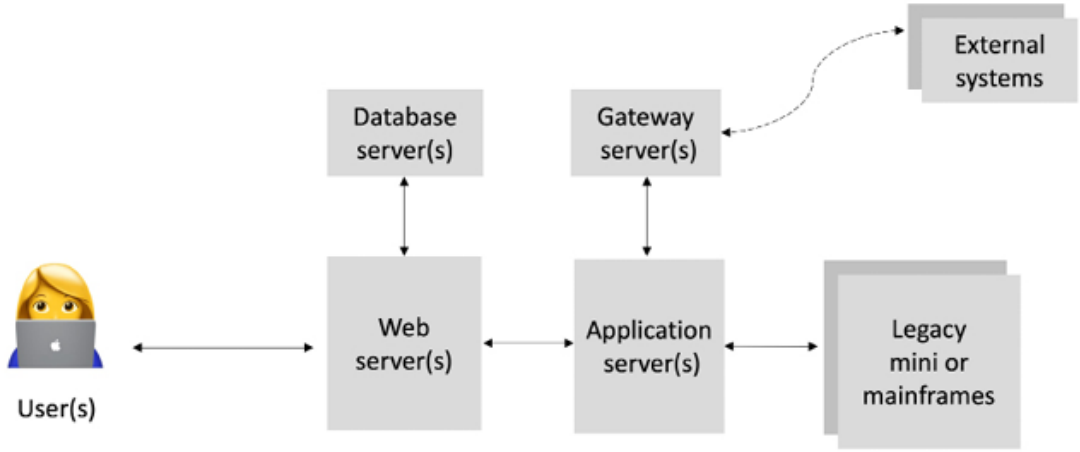

Figure 2.3: A traditional distributed system comprises many servers performing different roles

The following diagram shows a decentralized system (based on blockchain) where an exact replica of the applications and data is maintained across the entire network on each participating node:

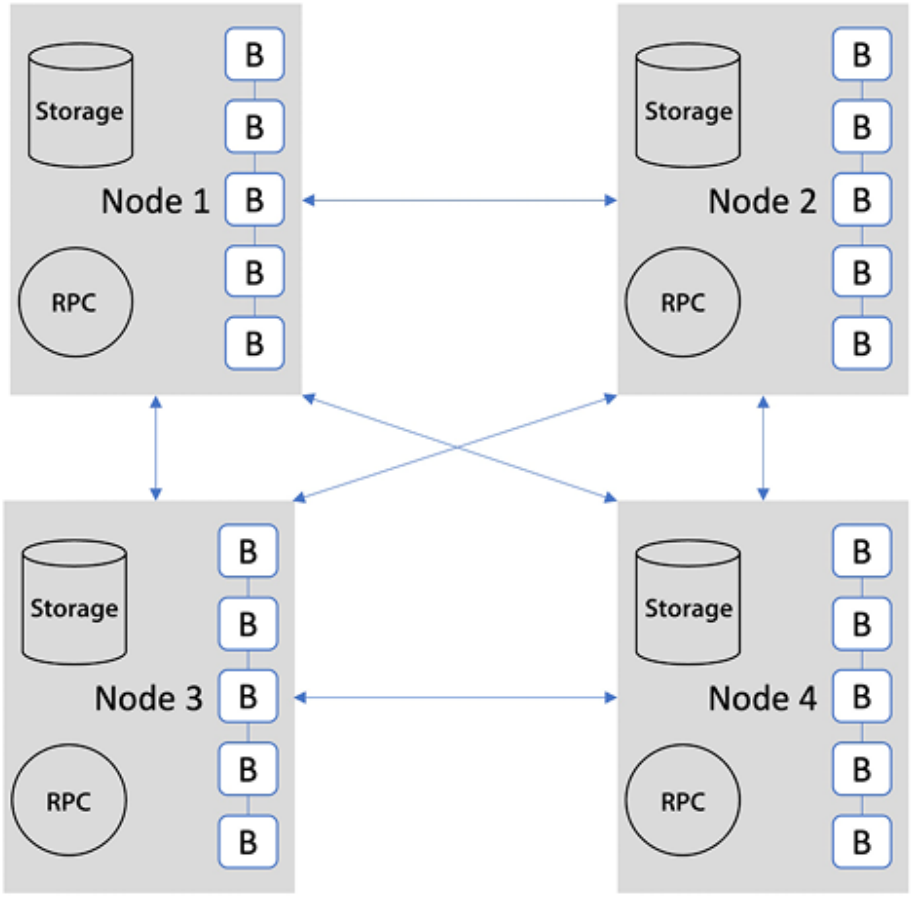

Figure 2.4: A blockchain-based decentralized system (notice the direct P2P connections and the exact replicas of blocks (data))

A comparison between centralized and decentralized systems is shown in the following table:

Feature	**Centralized**	**Decentralized**
Ownership	Service provider	All users
Architecture	Client/server	Distributed, different topologies

Security	Basic	More secure
High availability	No	Yes
Fault tolerance	Limited, single point of failure	Highly tolerant, as service is replicated
Collusion resistance	Basic, because it's under the control of a group or even a single individual	Highly resistant, as consensus algorithms ensure defense against adversaries
Application architecture	Single application	Application replicated across all nodes on the network
Trust	Consumers must trust the service provider, i.e., a trusted third party	No mutual trust required
Cost for consumer	High	Low

The comparison in the table only covers some main features and is not an exhaustive list, but this list should provide a good level of comparison.

Note that fault tolerance (the ability of a system to continue operating even if some of its components fail) in centralized systems is also improved by data replication. However, in the case of a decentralized system, fault tolerance is higher because, first, it's a distributed system, and secondly, it's decentralized, so no single participant could single-handedly game the system and gain a disproportionate advantage. If the architecture is a basic client/server with only one central server or perhaps just a primary and the backup server providing services, it will be considerably less fault-tolerant than a decentralized distributed blockchain system, as

blockchains are replicated across usually hundreds and thousands of replicas (participants) worldwide in different geographic locations.

Now we will discuss what methods can be used to achieve decentralization.

Methods of decentralization

Two methods can be used to achieve decentralization: disintermediation and competition. These methods will be discussed in detail in the sections that follow.

Disintermediation

The concept of **disintermediation** can be explained with an example. Imagine that you want to send money to a friend in another country. You go to a bank, which, for a fee, will transfer your money to the bank in that country. In this case, the bank maintains a central database that is updated, confirming that you have sent the money.

With blockchain technology, it is possible to send this money directly without the need for a bank. All you need is an address on the blockchain. This way, the intermediary (that is, the bank) is no longer required, and decentralization is achieved by disintermediation. It is debatable, however, how practical decentralization through disintermediation is in the financial sector due to the massive regulatory and compliance requirements.

> Central banks and monetary authorities have recognized that they can use blockchain to issue a regulated digital currency called **central bank digital currency** (**CBDC**), which can simplify the implementation and execution of monetary and fiscal policies. While this is a significant insight and could result in a safer, more efficient, and more financially inclusive financial ecosystem, it is expected to be centralized with a central bank or a nation's monetary authority regulating and issuing the currency.

Nevertheless, this model can be used not only in finance but in many other industries as well, such as health, law, and the public sector. In the health industry, instead of relying on a trusted third party (such as a hospital record system), patients can be in full control of their own identity and their data that they can share directly with only those entities that they trust. As a general solution, blockchain can serve as a decentralized health record management system where health records can be exchanged securely and directly between different entities (hospitals, pharmaceutical companies, patients) globally without any central authority. While interoperability between different standards for recording and categorizing health-related data is not easy, blockchain can at least provide a platform to share data between different health providers.

Contest-driven decentralization

In a method involving **competition**, different service providers compete to be selected for the provision of services by the system. This paradigm does not achieve complete decentralization. However, to a certain degree, it ensures that an intermediary or service provider is not monopolizing the service. In the context of blockchain technology, a system can be envisioned in which smart contracts can choose an external data provider from many providers based on their reputation, previous score, reviews, and quality of service. This method will not result in full decentralization, but it allows smart contracts to make a free choice based on various criteria. This way, an environment of competition is cultivated among service providers where they compete to become the service provider of choice.

Quantifying decentralization

In the following diagram, varying levels of decentralization are shown. On the left side, the conventional approach is shown where a central system is in control; on the right side, complete disintermediation is achieved, as intermediaries are entirely removed. Competing intermediaries or service providers are shown in the center.

At that level, intermediaries or service providers are selected based on reputation or voting, thus achieving partial decentralization:

Figure 2.5: Scale of decentralization

We can think about the decentralization spectrum from another angle, where the attainable decentralization ranges from **minimum achievable decentralization (MAD)** to **maximum feasible decentralization (MFD)**. Moreover, we can focus on finding an **optimal decentralization point (ODP)** on the spectrum for a given use case, that is, where the maximum decentralization is achieved along with as little centralization as possible, which is most favorable for the specific use case under consideration.

A question arises here regarding how we can measure the level of decentralization. An answer to this question is the "Nakamoto coefficient." This metric is calculated using several factors. It represents the number of entities that are required to be controlled to compromise a blockchain network. The higher the value of the Nakamoto coefficient, the more decentralized the network is.

> You can track the Nakamoto coefficient here: https://nakaflow.io/.

Any decentralized system is composed of several decentralized subsystems. If any subsystem is centralized, the overall system is considered centralized. For example, a blockchain can be composed of several subsystems, including miners, clients, developers, exchanges, nodes, and ownership. We can say that due to mining pools' monopoly on Bitcoin mining, Bitcoin can be considered centralized. The key insight behind this metric is to first enumerate the subsystems of a decentralized system, then figure out how many entities are required to be compromised to gain control of each subsystem, and then use the minimum of these values to get the overall effective decentralization of the system. For exam-

ple, Ethereum could be considered centralized because only a handful of developers do most of the commits, hence resulting in developer centralization.

> The Nakamoto coefficient was introduced by Balaji S. Srinivasan and Leland Lee in their article here: https://news.earn.com/quantifying-decentralization-e39db233c28e.

Benefits of decentralization

There are many benefits of decentralization, including transparency, efficiency, cost savings, the development of trusted ecosystems, and in some cases privacy and anonymity. Some challenges, such as security requirements, software bugs, and human error, need to be examined thoroughly.

For example, in a decentralized system such as Bitcoin or Ethereum where security is normally provided by private keys, how can we ensure that an asset or a token associated with these private keys cannot be rendered useless due to negligence or bugs in the code? What if the private keys are lost due to user negligence? What if due to a bug in the smart contract code the decentralized application becomes vulnerable to attack?

Before embarking on a journey to decentralize everything using blockchain and decentralized applications, it is essential that we understand that not everything can or needs to be decentralized.

This view raises some fundamental questions. Is a blockchain really needed? When is a blockchain required? In what circumstances is blockchain preferable to traditional databases? To answer these questions, go through the simple set of questions presented below:

Question	Yes/No	Recommended solution

Is high data throughput required?	Yes	Use a traditional database.
	No	A central database might still be useful if other requirements are met. For example, if users trust each other, then perhaps there is no need for a blockchain. However, if they don't or trust cannot be established for any reason, blockchain can be helpful.
Are updates centrally controlled?	Yes	Use a traditional database.
	No	You may investigate how a public/private blockchain can help.
Do users trust each other?	Yes	Use a traditional database.
	No	Use a public blockchain.
Are users anonymous?	Yes	Use a public blockchain.
	No	Use a private blockchain.
Is consensus required to be maintained within a consortium?	Yes	Use a private blockchain.
	No	Use a public blockchain.
Is strict data immutability required?	Yes	Use a blockchain.
	No	Use a central/traditional database.

Answering all these questions can help you decide whether a blockchain is required or suitable for solving a problem. Beyond the questions posed in this model, there are many other issues to consider, such as latency, the

choice of consensus mechanisms, whether consensus is required or not, and where consensus is going to be achieved.

If consensus is maintained internally by a consortium, then a private blockchain should be used; otherwise, if consensus is required publicly among multiple entities, then a public blockchain solution should be considered. Other aspects, such as immutability, should also be considered when deciding whether to use a blockchain or a traditional database. If strict data immutability is required, then a public blockchain should be used; otherwise, a central database may be an option.

As blockchain technology matures, there will be more questions raised regarding this selection model. For now, however, this set of questions is sufficient for deciding whether a blockchain-based solution is suitable or not.

Now we understand different methods of decentralization and have looked at how to decide whether a blockchain is required or not in a particular scenario. Let's now look at the process of decentralization, that is, how we can take an existing system and decentralize it.

Evaluating requirements

There are systems that pre-date blockchain and Bitcoin, including BitTorrent and the Gnutella file-sharing system, which to a certain degree could be classified as decentralized. However, due to a lack of any incentivization mechanism, participation from the community gradually decreased. With the advent of blockchain technology, many initiatives are being taken to leverage this new technology to achieve decentralization.

The Bitcoin blockchain has been typically the first choice for many, as it has proven to be the most resilient and secure blockchain. However, Ethereum has become a more prominent choice because of the flexibility it allows for programming any business logic into the blockchain by creating **smart contracts**. Moreover, newer chains like Polkadot, Solana, and Cardano are also used as platforms for decentralization by many developers.

The Nakamoto coefficient varies from chain to chain and should be a deciding factor during the evaluation of blockchain platforms for a use case. While it is best to be as decentralized as possible, in some cases depending on the use case, some decentralization can be given up. Bitcoin has the highest Nakamoto coefficient whereas some chains have a very low Nakamoto coefficient. The choice of blockchain platform is governed by the use case and user requirements and in some cases, even giving up some level of decentralization is acceptable.

Arvind Narayanan et al. have proposed a framework in their book *Bitcoin and Cryptocurrency Technologies* that can be used to evaluate the decentralization requirements of a variety of issues in the context of blockchain technology. The framework raises four questions whose answers provide a clear understanding of how a system can be decentralized:

- *What is being decentralized?:* This can be any system, such as an identity system or a trading system.
- *What level of decentralization is required?:* This can be full disintermediation or partial disintermediation.
- *What blockchain is used?:* It can be the Bitcoin blockchain, Ethereum blockchain, or any other blockchain that is deemed fit for the specific application.
- *What security mechanism is used?:* For example, the security mechanism can be atomicity-based, where either the transaction executes in full or does not execute at all. This deterministic approach ensures the integrity of the system. Other mechanisms include those based on reputation, which allows for varying degrees of trust in a system.

Next, let's evaluate a money transfer system as an example of an application selected to be decentralized. The four questions discussed previously are used to evaluate the decentralization requirements of this application. The answers to these questions are as follows:

- *What is being decentralized?:* A money transfer system.
- *What level of decentralization is required?:* Disintermediation.
- *What blockchain is used?:* Bitcoin.
- *What security mechanism is used?:* Atomicity.

The responses indicate that the money transfer system can be decentralized by removing the intermediary, implemented on the Bitcoin blockchain, and that a security guarantee can be provided via atomicity. Atomicity will ensure that transactions execute successfully in full or do not execute at all. We have chosen the Bitcoin blockchain because it is the longest-established blockchain and has stood the test of time.

Similarly, this framework can be used for any other system that needs to be evaluated in terms of decentralization. The answers to these four simple questions help clarify what approach to take to decentralize the system.

To achieve complete decentralization, it is necessary that the environment around the blockchain also be decentralized. We'll look at the full ecosystem of decentralization next.

Full-ecosystem decentralization

The blockchain is a distributed ledger that runs on top of conventional systems. These elements include storage, communication, and computation.

There are other factors, such as identity and wealth, which are traditionally based on centralized paradigms, and there's a need to decentralize these aspects as well to achieve a sufficiently decentralized ecosystem.

Storage

Data can be stored directly on a blockchain, and with this fact it achieves decentralization. However, a significant disadvantage of this approach is that a blockchain is not suitable for storing large amounts of data by design. It can store simple transactions and some arbitrary data, but it is certainly not suitable for storing images or large blobs of data, as is the case with traditional database systems.

A better alternative for storing data is to use **distributed hash tables (DHTs)**. DHTs were used initially in peer-to-peer file-sharing software, such as BitTorrent, Napster, Kazaa, and Gnutella. DHT research was made popular by the CAN, Chord, Pastry, and Tapestry projects. BitTorrent is

the most scalable and fastest network, but the issue with BitTorrent and the others is that there is no incentive for users to keep the files indefinitely.

Users generally don't keep files permanently, and if nodes that have data still required by someone leave the network, there is no way to retrieve that data except by having the required nodes rejoin the network so that the files once again become available.

Two primary requirements here are high availability, which means that data should be available when required, and link stability, meaning that network links also should always be accessible. **Inter-Planetary File System (IPFS)** possesses both properties, and its vision is to provide a decentralized World Wide Web by replacing the HTTP protocol. IPFS uses the Kademlia DHT and Merkle **Directed Acyclic Graphs (DAGs)** to provide its storage and searching functionality, respectively. The concept of DHTs and DAGs will be introduced in detail in *Chapter 4*, *Asymmetric Cryptography* and *Chapter 17*, *Scalability*, respectively.

The incentive mechanism for storing data is based on a protocol known as Filecoin, which pays incentives to nodes that store data using the Bitswap mechanism. The Bitswap mechanism lets nodes keep a simple ledger of bytes sent or bytes received in a one-to-one relationship. Also, a Git-based version control mechanism is used in IPFS to provide structure and control over the versioning of data.

There are other alternatives for data storage, such as Ethereum Swarm, Storj, and MaidSafe. Ethereum has its own decentralized and distributed ecosystem that uses Swarm for storage and the Whisper protocol for communication. MaidSafe aims to provide a decentralized World Wide Web. All these projects are discussed later in this book in greater detail.

BigChainDB is another storage layer decentralization project aimed at providing a scalable, fast, and linearly scalable decentralized database as opposed to a traditional filesystem. BigChainDB complements decentralized processing platforms and filesystems such as Ethereum and IPFS.

Communication

The Internet (the communication layer in blockchain) appears to be decentralized. This belief is correct to some extent, as the original vision of the Internet was to develop a decentralized communications system. Services such as email and online storage are now all based on a paradigm where the service provider is in control, and users trust such providers to grant them access to the service as requested. This model is based on the unconditional trust of a central authority (the service provider) where users are not in control of their data. Even user passwords are stored on trusted third-party systems.

Thus, there is a need to provide control to individual users in such a way that access to their data is guaranteed and is not dependent on a single third party. Access to the Internet is based on **Internet Service Providers (ISPs)** who act as a central hub for Internet users. If the ISP is shut down for any reason, then no communication is possible with this model.

An alternative is to use mesh networks. Mesh networks use wireless technologies such as **Bluetooth Low Energy (BLE)** to form communication networks that do not need Internet connectivity. Even though they are limited in functionality compared to the Internet, they still provide a decentralized alternative where nodes in relative proximity can talk directly to each other without requiring the Internet or a central hub such as an ISP. Now imagine a network that allows users to be in control of their communication; no one can shut it down for any reason. This type of network is very advantageous in situations like natural disasters or war zones. Another use could be to arrange protests against oppressive regimes that might have blocked the Internet. One example of such an offline messaging app is **bridgefy**. This could be the next step toward decentralizing communication networks in the blockchain ecosystem.

As mentioned earlier, the original vision of the Internet was to build a decentralized network; however, over the years, with the arrival of large-scale service providers such as Google, Amazon, and eBay, control has shifted toward these big players. For example, email is a decentralized system at its core; that is, anyone can run an email server with minimal effort and can start sending and receiving emails. However, there are better alternatives available. For example, Gmail and Outlook already provide managed services for end users, so there is a natural inclination to-

ward selecting one of these large, centralized services as they are more convenient, secure, and above all free to use. This is one example that shows how the Internet has moved toward centralization.

Free services, however, are offered at the cost of exposing valuable personal data, and many users are unaware of this fact. Blockchain has revived the vision of decentralization across the world, and now concerted efforts are being made to harness this technology and take advantage of the benefits that it can provide.

Computing power

The decentralization of computing or processing power is achieved by a blockchain technology such as Ethereum, where smart contracts with embedded business logic can run on the blockchain network. Other blockchain technologies also provide similar processing-layer platforms, where business logic can run over the network in a decentralized manner.

The following diagram shows an overview of a decentralized ecosystem:

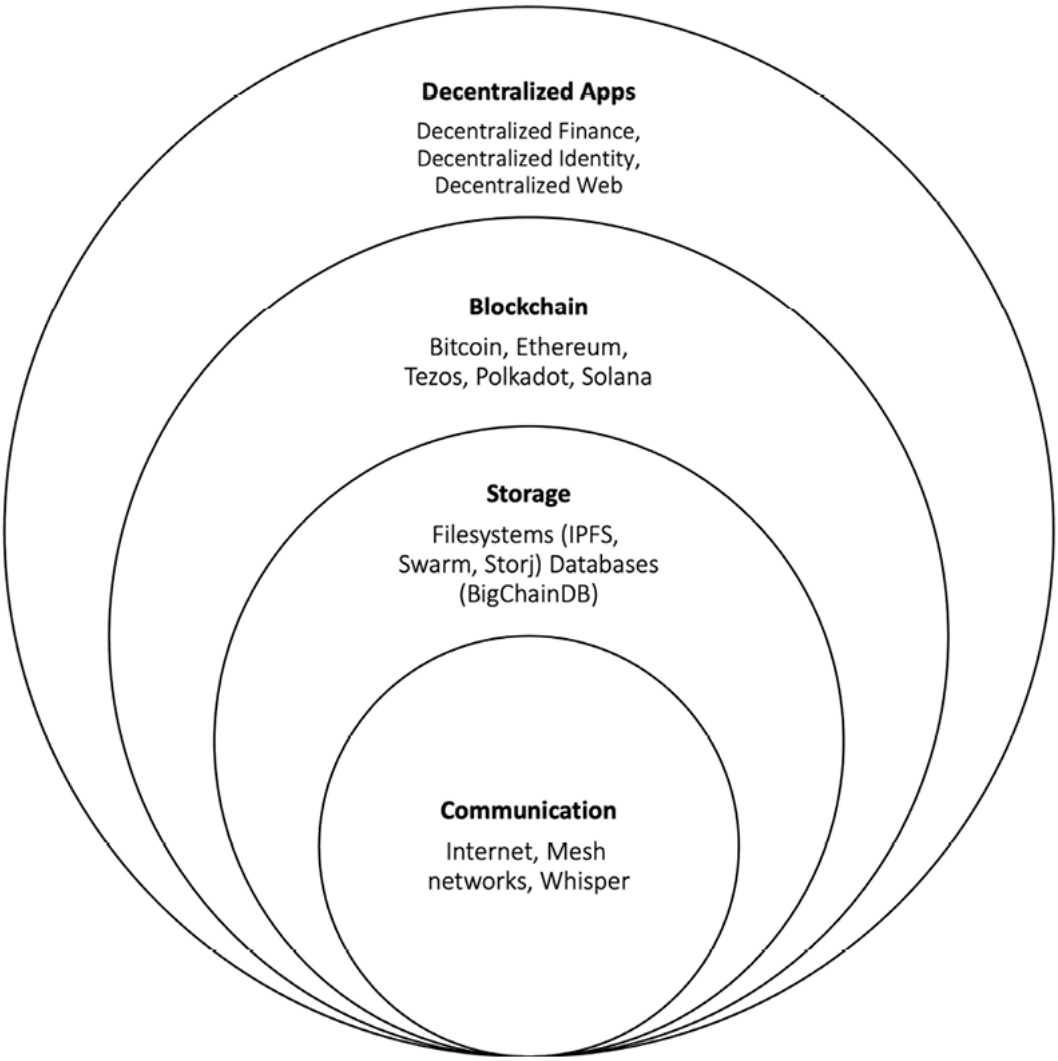

Figure 2.6: Decentralized ecosystem

In the bottom layer, the Internet or mesh networks provide a decentralized communication layer. In the next layer up, a storage layer uses technologies such as IPFS and BigChainDB to enable decentralization. Finally, in the next level up, you can see that the blockchain serves as a decentralized processing (computation) layer. Blockchain can, in a limited way, provide a storage layer too, but that severely hampers the speed and capacity of the system. Therefore, other solutions such as IPFS and BigChainDB are more suitable for storing large amounts of data in a decentralized way. The identity, finance, and web layers are shown at the top level.

The blockchain can provide solutions to various issues relating to decentralization. A concept relevant to identity known as **Zooko's triangle** requires that the naming system in a network protocol is secure, decentralized, and able to provide human-meaningful and memorable names to

the users. Conjecture has it that a system can have only two of these properties simultaneously.

Nevertheless, this problem was resolved with the advent of **Namecoin**, which makes it possible to achieve security, decentralization, and human-meaningful names. However, this is not a panacea, and it comes with many challenges, such as reliance on users to store and maintain private keys securely. This raises other general questions about the suitability of decentralization to a particular problem.

Decentralization may not be appropriate for every scenario. Centralized systems with well-established reputations tend to work better in many cases. For example, email platforms from reputable companies such as Google or Microsoft would provide a better service than a scenario where individual email servers were hosted privately by users on the Internet.

There are many projects underway that are developing solutions for a more comprehensive distributed blockchain system. For example, Swarm and Whisper are being developed to provide decentralized storage and communication for Ethereum.

With the advent of blockchain technology, it is now possible to build software versions of traditional physical organizations in the form of **Decentralized Organizations (DOs)** and other similar constructs, which we will examine in detail shortly.

Moreover, with the emergence of the decentralization paradigm, different terminology and buzzwords are now appearing in the media and academic literature, which we will explore in the next section.

Decentralization in practice

The following concepts are worth citing in the context of decentralization. The terminology introduced here is often used in the literature concerning decentralization and its applications.

Smart contracts

A **smart contract** is a software program that usually runs on a blockchain. Smart contracts do not necessarily need a blockchain to run; however, due to the security benefits that blockchain technology provides, blockchain has become a standard decentralized execution platform for smart contracts.

A smart contract usually contains some business logic and a limited amount of data. The business logic is executed if specific criteria are met. Actors or participants in the blockchain use these smart contracts, or they run autonomously on behalf of the network participants.

More information on smart contracts will be provided in *Chapter 8, Smart Contracts*.

Autonomous agents

An **Autonomous Agent** (**AA**) is a software entity (artificially intelligent or traditionally programmed) that acts on the behalf of its owner to achieve some desirable goals without requiring any or minimal intervention from its owner.

Decentralized organizations

DOs are software programs that run on a blockchain and are based on the model of real-life organizations with people and protocols. Once a DO is added to the blockchain in the form of a smart contract or a set of smart contracts, it becomes decentralized, and parties interact with each other based on the code defined within the DO software.

Decentralized autonomous organizations

Just like DOs, a **decentralized autonomous organization** (**DAO**) is also a computer program that runs on top of a blockchain, and embedded within it are governance and business logic rules. DAOs and DOs are fundamentally the same thing. The main difference, however, is that DAOs are autonomous, which means that they are fully automated and contain artificially intelligent logic. DOs, on the other hand, lack this feature and rely on human input to execute business logic.

The Ethereum blockchain led the way with the introduction of DAOs. In a DAO, the code is considered the governing entity rather than people or paper contracts. However, a human curator maintains this code and acts as a proposal evaluator for the community. DAOs can hire external contractors if enough input is received from the token holders (participants).

The most famous DAO project is **the DAO**, which raised $168 million in its crowdfunding phase. The DAO project was designed to be a venture capital fund aimed at providing a decentralized business model with no single entity as owner. Unfortunately, this project was hacked due to a bug in the DAO code, and millions of dollars' worth of **ether** currency (**ETH**) was siphoned out of the project and into a child DAO created by hackers. A major network change (hard fork) was required on the Ethereum blockchain to reverse the impact of the hack and initiate the recovery of the funds. This incident opened the debate on the security, quality, and need for thorough testing of the code in smart contracts to ensure their integrity and adequate control. There are other projects underway, especially in academia, that seek to formalize smart contract coding and testing.

Currently, DAOs do not have any legal status, even though they may contain some intelligent code that enforces certain protocols and conditions. However, these rules have no value in the real-world legal system at present. One day, perhaps an AA (that is, a piece of code that runs without human intervention) commissioned by a law enforcement agency or regulator will contain rules and regulations that could be embedded in a DAO for the purpose of ensuring its integrity from a legalistic and compliance perspective. The fact that DAOs are purely decentralized entities enables them to run in any jurisdiction. Thus, they raise a big question as to how the current legal system could be applied to such a varied mix of jurisdictions and geographies.

Decentralized autonomous corporations

A **decentralized autonomous corporation** (**DAC**) is like a DAO in concept, though considered to be a subset of them. The definitions of DACs and DAOs may sometimes overlap, but the general distinction is that DAOs are usually considered to be nonprofit, whereas DACs can earn a

profit via shares offered to the participants and to whom they can pay dividends. DACs can run a business automatically without human intervention based on the logic programmed into them.

Decentralized autonomous societies

A **decentralized autonomous society (DAS)** is an entire society that can function on a blockchain with the help of multiple, complex smart contracts and a combination of DAOs and **decentralized applications (DApps)** running autonomously. This model does not necessarily translate to a free-for-all approach, nor is it based on an entirely libertarian ideology; instead, many services that a government commonly offers can be delivered via blockchains, such as government identity card systems, passports, and records of deeds, marriages, and births. Another theory is that, if a government is corrupt and central systems do not provide the levels of trust that a society needs, then that society can start its own virtual one on a blockchain that is driven by decentralized consensus and transparency. This concept might look like a libertarian's or cypherpunk's dream, but it is entirely possible on a blockchain.

This concept also goes into the realm of algocracy, an alternative form of governance and social system where computer algorithms maintain, control, and automate public services such as law, legal system, regulation, governance, economics, policies, and public decision-making. Blockchain and DApps are well-suited means to enable algocracy, especially when combined with AI. Initially, AI (or even traditionally programmed software) was seen to allow algorithmic governance; now, blockchain combined with AI can offer a more elegant approach.

However, there are both opportunities and threats associated with Algocracy. Increasing reliance on governance by algorithms is seen as a threat to active human participation and real-life decision-making. This is true in traditional algocracy without blockchain. However, when combined with blockchain, the situation improves. Due to the blockchain's decentralized and community-governed model, governance algorithms are also subject to approval and scrutiny by the community (society) operating on the blockchain. Therefore, blockchain can be seen as a solution to this threat of losing control of the decision-making process. We could call

this variation of algocracy “blockcracy”, i.e., government by blockchain, after the vision of a blockchain running artificially intelligent smart contracts (algorithms) responsible for governance.

Decentralized applications

All the ideas mentioned up to this point come under the broader umbrella of decentralized applications, abbreviated to DApps (pronounced Dee-App, or now more commonly rhyming with app). DAOs, DACs, and DOs are DApps that run on top of a blockchain in a peer-to-peer network. They represent the latest advancement in decentralization technology.

DApps at a fundamental level are software programs that execute using either of the following methods. They are categorized as Type 1, Type 2, or Type 3 DApps:

- **Type 1**: These run on their own dedicated blockchain, for example, standard smart contract-based DApps running on Ethereum. If required, they make use of a native token, for example, ETH on the Ethereum blockchain. For example, **Ethlance** is a DApp that makes use of ETH to provide a job market.

More information about Ethlance can be found at `https://ethlance.com`.

- **Type 2**: These use an existing established blockchain. That is, they make use of Type 1 blockchain and bear custom protocols and tokens, for example, smart contract-based tokenization DApps running on the Ethereum blockchain. An example is **DAI**, which is built on top of the Ethereum blockchain, but contains its own stablecoins and mechanism of distribution and control. Another example is **Golem**, which has its own token GNT and a transaction framework built on top of the Ethereum blockchain to provide a **decentralized marketplace** for computing power where users share their computing power with each other in a peer-to-peer network. An example of Type 2 DApps is the OMNI network, which is a software layer built on top of Bitcoin to support the trading of custom digital assets and digital currencies.

> More information on the OMNI network can be found at `https://www.omnilayer.org`. More information on the Golem network is available at `https://golem.network`. More information on DAI is available at `https://makerdao.com/en/`.

- **Type 3**: Use the protocols of Type 2 DApps; for example, the SAFE Network uses the OMNI network protocol.

> More information on the SAFE Network can be found at `https://safenetwork.tech`.

Another example to understand the difference between different types of DApps is the USDT token (Tether). The original USDT uses the OMNI layer (a Type 2 DApp) on top of the Bitcoin network. USDT is also available on Ethereum using ERC-20 tokens. This example shows that USDT can be considered a Type 3 DApp, where the OMNI layer protocol (a Type 2 DApp) is used, which is itself built on Bitcoin (a Type 1 DApp). Also, from the perspective of Ethereum, USDT can also be considered a Type 3 DApp in that it makes use of the Type 1 DApp Ethereum blockchain using the ERC-20 standard, which was built to operate on Ethereum.

> More information can be found about Tether at `https://tether.to`.

In the last few years, the expression DApp has been increasingly used to refer to any end-to-end decentralized blockchain application, including a user interface (usually a web interface), smart contract(s), and the host blockchain. The clear distinction between different types of DApps is now not commonly referred to, but it does exist. Often, no reference to their type is made and they are all called just DApps.

There are thousands of different DApps running on various platforms (blockchains) now. There are various categories of these DApps covering media, social, finance, games, insurance, and health. There are various decentralized platforms (or blockchains), such as Ethereum, Solana,

Avalanche, Polkadot, and EOS. Some DApps stats are available here: https://thedapplist.com.

Criteria for a DApp

For an application to be considered decentralized, it should meet the following criteria:

- **Decentralized**: The DApp should be fully decentralized. In other words, no single entity should be in control of its operations. All changes to the application must be consensus-driven based on the view given by the community.
- **Opensource**: It must be open source for public scrutiny and transparency.
- **Cryptographically secure**: State transition and the data of the application must be cryptographically secured and stored on a blockchain to avoid any central points of failure. Note that data does not necessarily need to be encrypted to provide confidentiality but should be protected against unauthorized manipulation. Data integrity, authentication, and non-repudiation services should be provided.
- **Incentive availability**: A cryptographic token must be used by the application to provide access and incentives for those who contribute value to the applications, for example, miners in Bitcoin. This requirement can be relaxed in a consortium chain where a token can still be used for value transfer, but not as a cryptocurrency.
- **Proof of value**: The tokens (if applicable) must be generated by the decentralized application using consensus and an applicable cryptographic algorithm. This generation of tokens acts as a proof of the value to contributors (for example, miners).

Generally, DApps now provide all sorts of different services, including but not limited to financial applications, gaming, social media, and supply chain management.

Operations of a DApp

Establishment of consensus by a DApp can be achieved using consensus algorithms such as **Proof of Work** (**PoW**) and **Proof of Stake** (**PoS**). So far, only PoW has been found to be incredibly resistant to attacks, as is

evident from the trust people have put in the Bitcoin network, along with its success. Furthermore, a DApp can distribute tokens (coins) via **mining**, **fundraising**, and **development**.

Design of a DApp

A DApp is a software application that runs on a decentralized network such as a distributed ledger. They have recently become very popular due to the development of various decentralized platforms such as Ethereum, Solana, EOS, and Tezos.

Traditional apps commonly consist of a user interface and usually a web server or an application server and a backend database. This is a common client/server architecture. This is visualized in the following diagram:

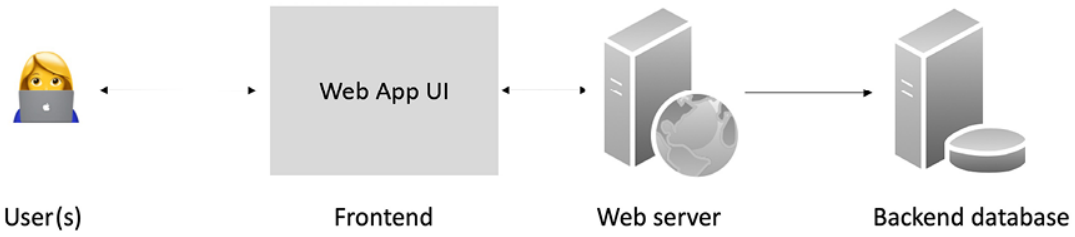

Figure 2.7: Traditional application architecture (generic client/server)

A DApp on the other hand has a blockchain as a backend and can be visualized as depicted in the following diagram. The key element that plays a vital role in the creation of a DApp is a smart contract that runs on the blockchain and has business logic embedded within it:

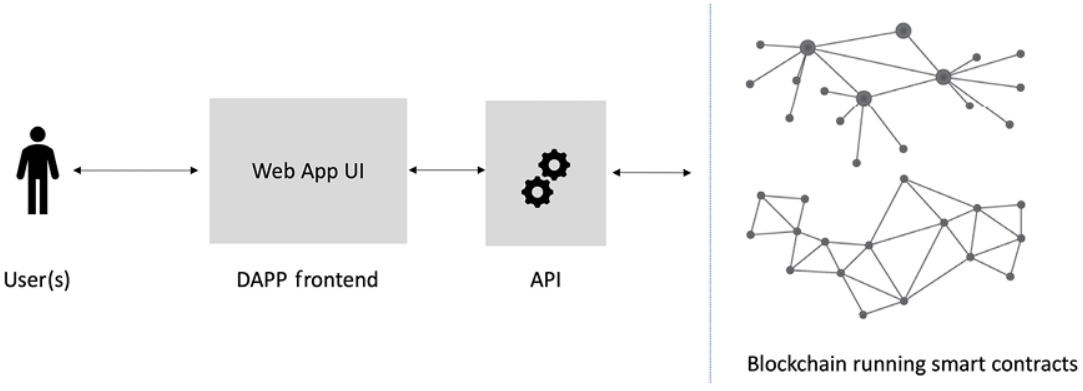

Figure 2.8: Generic DApp architecture

Note that the frontend in a DApp can either be a thick client, a mobile app, or a web frontend (a web user interface). However, it is usually a web frontend commonly written using a JavaScript framework such as React or Angular.

The following comparison table highlights the key properties of and differences between these different types of decentralized entities:

Entity	Autonomous?	Software?	Owned?	Capital?	Legal status?	Cost
DO	No	No	Yes	Yes	Yes	High
DAO	Yes	Yes	No	Yes	Some work has begun	Low
DAC	Yes	Yes	Yes	Yes	Unsettled	Low
DAS	Yes	Yes	No	Possible	Unsettled	Low
DApp	Yes	Yes	Yes	Optional tokens	Unsettled	Use case dependent

It is expected that all these entities will be regulated and will have some legal standing in the future while remaining decentralized.

> For example, see the following article regarding DAO legality: https://fedsoc.org/commentary/fedsoc-blog/the-legal-status-of-decentralized-autonomous-organizations-do-daos-require-new-business-structures-some-states-think-so.

Any blockchain network, such as Bitcoin, Ethereum, Solana, Hyperledger Fabric, or Quorum, can serve as a decentralization platform on which DApps can be built and hosted.

Due to fast-paced innovation and the evolution of blockchain, many innovative trends have emerged, which we explore in the next section.

Innovative trends

With the growth of blockchain, several ideas have emerged that make use of the decentralization aspect of blockchain to provide more user-centric and fully decentralized services. Some of the key ideas in this space are the decentralized web, decentralized identity, and decentralized finance.

Decentralized web

Decentralized web is a term that's used to describe a vision of the web where no central authority or set of authorities will be in control. The original intention of the Internet was indeed decentralized, and the development of open protocols such HTTP, SMTP, and DNS meant that any individual could use these protocols freely, and immediately become part of the Internet. This is still true; however, with the emergence of a layer above these protocols called the **Web layer** a more service-oriented infrastructure was introduced, which inevitably led to large profit-seeking companies taking over. This is evident from the rise of Facebook, Google, Twitter, and Amazon, which of course provide excellent user services but at the cost of a more controlled, centralized, and closed system.

Once intended and developed as decentralized, open and free protocols are now being dominated by powerful commercial entities around the world, which has resulted in major concerns around privacy and data protection. These types of business models do work well and are quite popular due to the high level of standardization and services provided, but they pose a threat to privacy and decentralization due to the dominance of only a handful of entities on the entire Internet.

With blockchain, it is envisioned that this situation will change as it will allow development of the decentralized Internet, or the decentralized web, or Web 3 for short, which was the original intention of the Internet.

We can review the evolution of the Web over the last few decades by dividing the major developments into three key stages, Web 1, Web 2, and Web 3.

Web 1

This is the original World Wide Web, which was developed in 1989. This was the era when static web pages were hosted on servers and usually only allowed read actions from a user's point of view.

Web 2

This is the era when more service-oriented and web-hosted applications started to emerge around 2003. E-commerce websites, social networking, social media, blogs, multimedia sharing, mashups, and web applications are the main features of this period. The current web is Web 2, and even though we have a richer and more interactive Internet, all these services are still centralized. Web 2 has generated massive economic value and provides services that are essential for day-to-day business, personal use, social interactions, and almost every walk of life, but privacy concerns, the need for trusted third parties, and data breaches are genuine issues that need to be addressed. Common examples of centralized Web 2 services include Twitter, Facebook, Google Docs, and email services such Gmail and Hotmail.

Web 3

This is the vision of the decentralized internet or web that will revolutionize the way we use the internet today. This is the era that will be fully user-centric and decentralized without any single authority, large organization, or internet company in control. Some examples of Web 3 are as follows:

- **Steemit**: This is a social media platform based on the Steem blockchain and STEEM cryptocurrency. This cryptocurrency is awarded to contributors for the content they have shared, and the more votes they get, the more tokens they earn. More information is available at **https://steemit.com**.

- **Status**: This is a decentralized multipurpose communication platform providing secure and private communication. More information is available at `https://status.im`.
- **IPFS**: This is a peer-to-peer hypermedia/storage protocol that allows the storage and sharing of data in a decentralized fashion across a peer-to-peer network. More information is available at `https://ipfs.io`.

Other examples include OpenSea, a marketplace for trading NFTs, UniSwap, a decentralized cryptocurrency exchange, and Augur, a decentralized exchange. In Web 3, 3D virtual worlds called metaverses are likely to be extensively used.

Other fast-growing and exciting applications include **decentralized identity and decentralized finance** (**DeFi**), which we introduce in later chapters of this book.

Summary

This chapter introduced the concept of decentralization, which is the core service offered by blockchain technology. Although the concept of decentralization is not new, it has gained renewed significance in the world of blockchain. As such, various applications based on a decentralized architecture have recently been introduced.

The chapter began with an introduction to the concept of decentralization. Next, decentralization from the blockchain perspective was discussed. Moreover, ideas relating to the different layers of decentralization in the blockchain ecosystem were introduced. Several new concepts and terms have emerged with the advent of blockchain technology and decentralization from the blockchain perspective, including DAOs, DACs, and DApps. Finally, some innovative trends relating to DApps were presented. We also touched upon the concepts of algocracy and blockcracy.

In the next chapter, fundamental concepts necessary to understand blockchain technology will be presented—principally cryptography, which provides a crucial foundation for blockchain technology.

Join us on Discord!

To join the Discord community for this book – where you can share feedback, ask questions to the author, and learn about new releases – follow the QR code below:

https://packt.link/ips2H