Blockchain 101 1

If you are reading this book, it is very likely that you already have heard about blockchain and have some fundamental appreciation of its enormous potential. If not, then let me tell you that this is a technology that has promised to positively alter the existing paradigms of nearly all industries including, but not limited to IT, finance, government, media, medical, and law.

This chapter serves an introduction to blockchain technology, its technical foundations, the theory behind it, and various techniques that have been combined together to build what is known today as blockchain.
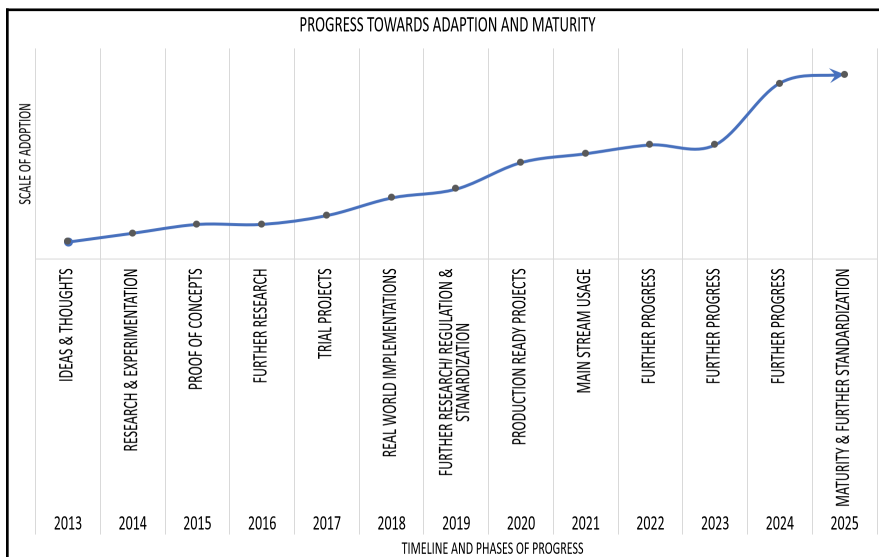
In this chapter, we first describe the theoretical foundations of distributed systems. Next, we address the precursors of Bitcoin by which blockchain technology was introduced to the world. Finally, we introduce you to blockchain technology. This approach is a logical way to understanding blockchain technology, as the roots of blockchain are in distributed systems. We will cover a lot of ground quickly here, but don't worry—we will go over a great deal of this material in much greater detail as you move through the book.

The growth of blockchain technology

With the invention of Bitcoin in 2008, the world was introduced to a new concept, which is now likely to revolutionize the whole of society. It is something that promises to have an impact on every industry, including but not limited to the financial sector, government, media, law, and arts. Some describe blockchain as a revolution, whereas another school of thought believes that it is going to be more evolutionary, and it will take many years before any practical benefits of blockchain reach fruition. This thinking is correct to some extent, but in my opinion, the revolution has already begun.

Many prominent organizations all around the world are already writing proofs of concept using blockchain technology, as its disruptive potential has now been fully recognized. However, some organizations are still in the preliminary exploration stage, though they are expected to progress more quickly as the technology matures. It is a technology that has an impact on current technologies too and possesses the ability to change them at a fundamental level.

If we look at the last few years, we notice that in 2013 some ideas started to emerge that suggested usage of blockchain in other areas than cryptocurrencies. Around that time the primary usage of blockchain was cryptocurrencies, and many new coins emerged during that time. The following graph shows a broad-spectrum outline of year wise progression and adaption trend of blockchain technology. Years shown on the *x* axis indicate the range of time in which a specific phase of blockchain technology falls. Each phase has a name which represents the action and is shown on the *x* axis starting from the period of **IDEAS & THOUGHTS** to eventually **MATURITY & FURTHER STANDARDIZATION**. The *y* axis shows level of activity, involvement and adoption of blockchain technology. The graph shows that eventually, roughly around **2025** blockchain technology is expected to become mature with a high number of users.

Blockchain technology adoption and maturity

The preceding graph shows that in 2013 **IDEAS & THOUGHTS** emerged related to other usages of blockchain technology apart from cryptocurrencies. Then in 2014 some **RESEARCH & EXPERIMENTATION** started which led to **PROOF OF CONCEPTS**, **FURTHER RESEARCH**, and full-scale **TRIAL PROJECTS** between 2015 and 2017. In 2018 we will see **REAL WORLD IMPLEMENTATIONS**. Already many projects are underway and set to replace existing systems, for example, **Australian Securities Exchange** (**ASX**) is soon to become the first organization to replace its legacy clearing and settlement system with blockchain technology.

More information on this topic can be found at `https://www.asx.com.au/services/chess-replacement.htm`.

It is expected that during 2019 more research will be carried out along with some interest towards regulation and standardization of blockchain technology. After this, production ready projects and off the shelf products utilizing blockchain technology will be available from 2020 and by 2021 mainstream usage of blockchain technology is expected to start. Progress in blockchain technology almost feels like the internet *dot-com boom* of the late 1990s. More research is expected to continue along with adaption and further maturity of blockchain technology, and finally, in 2025 it is expected that the technology will be mature enough to be used on day to day basis. Please note that the timelines provided in the chart are not strict and can vary as it is quite difficult to predict that when exactly blockchain technology will become mature. This graph is based on the progress made in the recent years and the current climate of research, interest and enthusiasm regarding this technology which suggests that by 2025 blockchain technology is expected to become mature.

Interest in blockchain technology has risen quite significantly over the last few years. Once dismissed as simply geek money from a cryptocurrency point of view, or as something that was just not considered worth pursuing, blockchain is now being researched by the largest companies and organizations around the world. Millions of dollars are being spent to adapt and experiment with this technology. This is evident from recent actions taken by European Union where they have announced plans to increase funding for blockchain research to almost 340 million euros by 2020.

Interested readers can read more about this at `https://www.irishtimes.com/business/technology/boost-for-blockchain-research-as-eu-increases-funding-four-fold-1.3383340`.

Another report suggests that global spending on blockchain technology research could reach 9.2 billion dollars by 2021.

> More information regarding this can be found at
> `https://bitcoinmagazine.com/articles/report-suggests-global-spen`
> `ding-blockchain-tech-could-reach-92-billion-2021/`.

There are various consortiums such as **Enterprise Ethereum Alliance** (**EEA**), **Hyperledger**, and **R3**, which have been established for research and development of blockchain technology. Moreover, a large number of start-ups are providing blockchain-based solutions already. A simple trend search on Google reveals the immense scale of interest in blockchain technology over the last few years. Especially, since early 2017 the increase in the search term *blockchain* is quite significant, as shown in the following graph:

Google trend graph for blockchain

Various benefits of this technology are envisioned, such as decentralized trust, cost savings, transparency, and efficiency. However, there are multiple challenges too that are an area of active research on blockchain, such as scalability and privacy.

In this book, we are going to see how blockchain technology can help bring about the benefits mentioned earlier. You are going to learn about what exactly is blockchain technology, and how it can reshape businesses, multiple industries, and indeed everyday life by bringing about a plenitude of benefits such as efficiency, cost saving, transparency, and security. We will also explore what is distributed ledger technology, decentralization, and smart contracts and how technology solutions can be developed and implemented using mainstream blockchain platforms such as Ethereum, and Hyperledger. We will also investigate that what challenges need to be addressed before blockchain can become a mainstream technology.

`Chapter 18`, *Scalability and Other Challenges*, is dedicated to a discussion of the limitations and challenges of blockchain technology.

Distributed systems

Understanding distributed systems is essential to the understanding of blockchain technology, as blockchain is a distributed system at its core. It is a distributed ledger which can be centralized or decentralized. A blockchain is originally intended to be and is usually used as a decentralized platform. It can be thought of as a system that has properties of both decentralized and distributed paradigms. It is a decentralized-distributed system.

Distributed systems are a computing paradigm whereby two or more nodes work with each other in a coordinated fashion to achieve a common outcome. It is modeled in such a way that end users see it as a single logical platform. For example, Google's search engine is based on a large distributed system, but to a user, it looks like a single, coherent platform.

A **node** can be defined as an individual player in a distributed system. All nodes are capable of sending and receiving messages to and from each other. Nodes can be honest, faulty, or malicious, and they have memory and a processor. A node that exhibits irrational behavior is also known as a **Byzantine node** after the Byzantine Generals Problem.
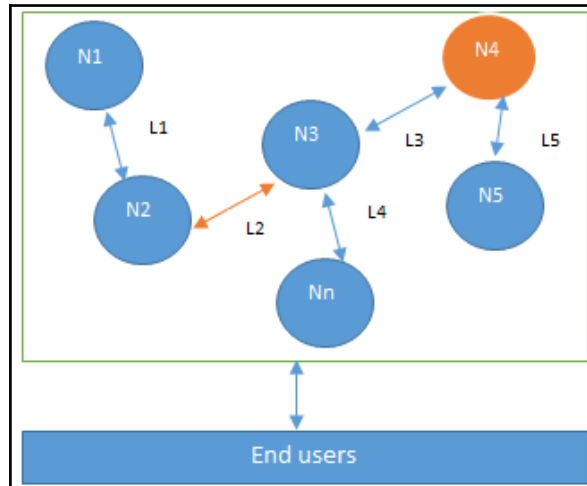
The Byzantine Generals problem

In 1982, a thought experiment was proposed by Lamport and others in their research paper, *The Byzantine Generals Problem* which is available at: `https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/` whereby a group of army generals who lead different parts of the Byzantine army are planning to attack or retreat from a city. The only way of communicating among them is via a messenger. They need to agree to strike at the same time in order to win. The issue is that one or more generals might be traitors who could send a misleading message. Therefore, there is a need for a viable mechanism that allows for agreement among the generals, even in the presence of the treacherous ones, so that the attack can still take place at the same time. As an analogy to distributed systems, the generals can be considered nodes, the traitors as Byzantine (malicious) nodes, and the messenger can be thought of as a channel of communication among the generals.

This problem was solved in 1999 by Castro and Liskov who presented the **Practical Byzantine Fault Tolerance** (**PBFT**) algorithm, where consensus is reached after a certain number of messages are received containing the same signed content.

This type of inconsistent behavior of Byzantine nodes can be intentionally malicious, which is detrimental to the operation of the network. Any unexpected behavior by a node on the network, whether malicious or not, can be categorized as Byzantine.

A small-scale example of a distributed system is shown in the following diagram. This distributed system has six nodes out of which one (**N4**) is a Byzantine node leading to possible data inconsistency. **L2** is a link that is broken or slow, and this can lead to partition in the network.

Design of a distributed system: N4 is a Byzantine node, L2 is broken or a slow network link

The primary challenge in distributed system design is coordination between nodes and fault tolerance. Even if some of the nodes become faulty or network links break, the distributed system should be able to tolerate this and continue to work to achieve the desired result. This problem has been an active area of distributed system design research for many years, and several algorithms and mechanisms have been proposed to overcome these issues.

Distributed systems are so challenging to design that a hypothesis known as the **CAP theorem** has been proven, which states that a distributed system cannot have all three of the much-desired properties simultaneously; that is, consistency, availability, and partition tolerance. We will dive into the CAP theorem in more detail later in this chapter.

The history of blockchain and Bitcoin

Blockchain was introduced with the invention of Bitcoin in 2008. Its practical implementation then occurred in 2009. For the purposes of this chapter, it is sufficient to review Bitcoin very briefly, as it will be explored in great depth in `Chapter 5`, *Introducing Bitcoin*. However, it is essential to refer to Bitcoin because, without it, the history of blockchain is not complete.

Electronic cash

The concept of electronic cash or digital currency is not new. Since the 1980s, e-cash protocols have existed that are based on a model proposed by David Chaum.

Just as understanding the concept of distributed systems is necessary to comprehend blockchain technology, the idea of electronic cash is also essential in order to appreciate the first and astonishingly successful application of blockchain, Bitcoin, or more broadly cryptocurrencies in general.

Two fundamental e-cash system issues need to be addressed: accountability and anonymity.
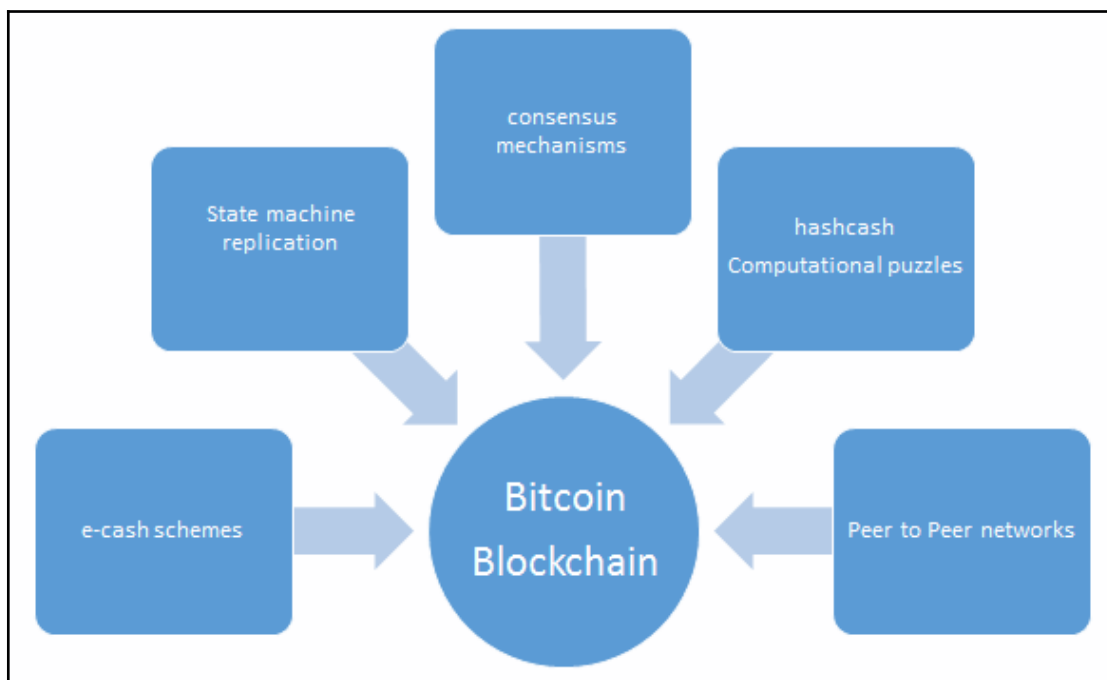
Accountability is required to ensure that cash is spendable only once (double-spend problem) and that it can only be spent by its rightful owner. Double spend problem arises when same money can be spent twice. As it is quite easy to make copies of digital data, this becomes a big issue in digital currencies as you can make many copies of same digital cash. **Anonymity** is required to protect users' privacy. As with physical cash, it is almost impossible to trace back spending to the individual who actually paid the money.

David Chaum solved both of these problems during his work in 1980s by using two cryptographic operations, namely **blind signatures** and **secret sharing**. These terminologies and related concepts will be discussed in detail in `Chapter 3`, *Symmetric Cryptography* and `Chapter 4`, *Public Key Cryptography*. For the moment, it is sufficient to say that *blind signatures* allow for signing a document without actually seeing it, and *secret sharing* is a concept that enables the detection of double spending, that is using the same e-cash token twice (double spending).

In 2009, the first practical implementation of an electronic cash (e-cash) system named Bitcoin appeared. The term cryptocurrency emerged later. For the very first time, it solved the problem of distributed consensus in a trustless network. It used **public key cryptography** with a **Proof of Work** (**PoW**) mechanism to provide a secure, controlled, and decentralized method of minting digital currency. The key innovation was the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism. This concept will be explained in greater detail in `Chapter 5`, *Introducing Bitcoin*.

Other technologies used in Bitcoin, but which existed before its invention, include Merkle trees, hash functions, and hash chains. All these concepts are explained in appropriate depth in `Chapter 4`, *Public Key Cryptography*.

Looking at all the technologies mentioned earlier and their relevant history, it is easy to see how concepts from electronic cash schemes and distributed systems were combined to create Bitcoin and what now is known as blockchain. This concept can also be visualized with the help of the following diagram:

The various ideas that supported the invention of Bitcoin and blockchain

Blockchain

In 2008, a groundbreaking paper entitled *Bitcoin: A Peer-to-Peer Electronic Cash System* was written on the topic of peer-to-peer electronic cash under the pseudonym *Satoshi Nakamoto*. It introduced the term **chain of blocks**. No one knows the actual identity of Satoshi Nakamoto. After introducing Bitcoin in 2009, he remained active in the Bitcoin developer community until 2011. He then handed over Bitcoin development to its core developers and simply disappeared. Since then, there has been no communication from him whatsoever, and his existence and identity are shrouded in mystery. The term *chain of blocks* evolved over the years into the word *blockchain*.

As stated earlier, blockchain technology incorporates a multitude of applications that can be implemented in various economic sectors. Particularly in the finance sector, significant improvement in the performance of financial transactions and settlements is seen as resulting in desirable time and cost reductions. Additional light will be shed on these aspects of blockchain in `Chapter 17`, *Blockchain – Outside of Currencies* where practical use cases will be discussed in detail for various industries. For now, it is sufficient to say that parts of nearly all economic sectors have already realized the potential and promise of blockchain and have embarked, or will do so soon, on the journey to capitalize on the benefits of blockchain technology.

Blockchain defined

Layman's definition: Blockchain is an ever-growing, secure, shared record keeping system in which each user of the data holds a copy of the records, which can only be updated if all parties involved in a transaction agree to update.

Technical definition: Blockchain is a peer-to-peer, distributed ledger that is cryptographically-secure, append-only, immutable (extremely hard to change), and updateable only via consensus or agreement among peers.

Now let's examine the preceding definitions in more detail. We will look at all keywords in the definitions one by one.

Peer-to-peer

The first keyword in the technical definition is *peer-to-peer*. This means that there is no central controller in the network, and all participants talk to each other directly. This property allows for cash transactions to be exchanged directly among the peers without a third-party involvement, such as by a bank.

Distributed ledger

Dissecting the technical definition further reveals that blockchain is a *distributed ledger*, which simply means that a ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger.

Cryptographically-secure

Next, we see that this ledger is *cryptographically-secure*, which means that cryptography has been used to provide security services which make this ledger secure against tampering and misuse. These services include non-repudiation, data integrity, and data origin authentication. You will see how this is achieved later in `Chapter 3`, *Symmetric Cryptography* which introduces the fascinating world of cryptography.
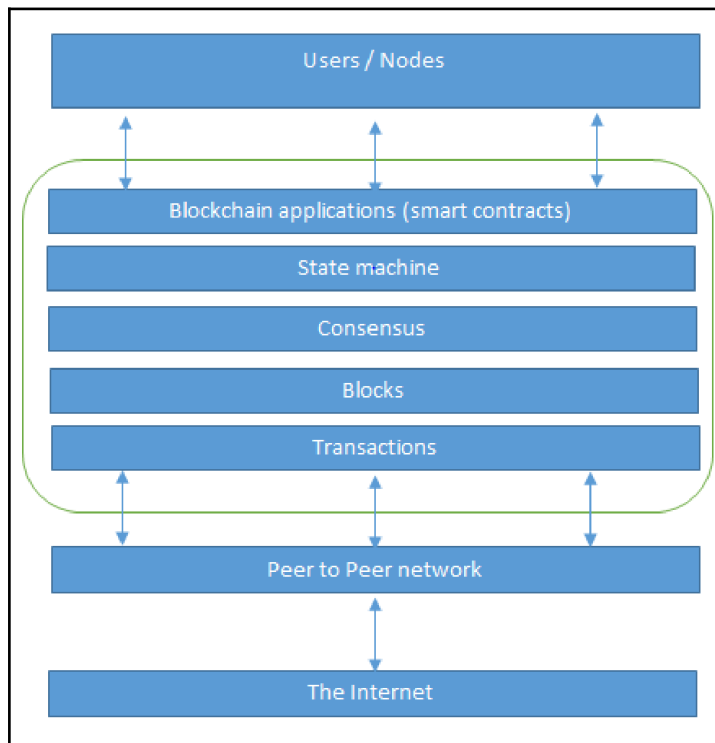
Append-only

Another property that we encounter is that blockchain is *append-only*, which means that data can only be added to the blockchain in *time-ordered sequential order*. This property implies that once data is added to the blockchain, it is almost impossible to change that data and can be considered practically immutable. Nonetheless, it can be changed in rare scenarios wherein collusion against the blockchain network succeeds in gaining more than 51 percent of the power. There may be some legitimate reasons to change data in the blockchain once it has been added, such as the *right to be forgotten* or *right to erasure* (also defined in **General Data Protection** (**GDPR**) ruling, `https://gdpr-info.eu/art-17-gdpr/`).

However, those are individual cases that need to be handled separately and that require an elegant technical solution. For all practical purposes, blockchain is indeed immutable and cannot be changed.

Updateable via consensus

Finally, the most critical attribute of a blockchain is that it is *updateable* only via consensus. This is what gives it the power of decentralization. In this scenario, no central authority is in control of updating the ledger. Instead, any update made to the blockchain is validated against strict criteria defined by the blockchain protocol and added to the blockchain only after a consensus has been reached among all participating peers/nodes on the network. To achieve consensus, there are various consensus facilitation algorithms which ensure that all parties are in agreement about the final state of the data on the blockchain network and resolutely agree upon it to be true. Consensus algorithms are discussed later in this chapter and throughout the book as appropriate.

Blockchain can be thought of as a layer of a distributed peer-to-peer network running on top of the internet, as can be seen in the following diagram. It is analogous to SMTP, HTTP, or FTP running on top of TCP/IP.

The network view of a blockchain

At the bottom layer in the preceding diagram, there is the internet, which provides a basic communication layer for any network. In this case, a peer-to-peer network runs on top of the internet, which hosts another layer of blockchain. That layer contains transactions, blocks, consensus mechanisms, state machines, and blockchain smart contracts. All of these components are shown as a single logical entity in a box, representing blockchain above the peer-to-peer network. Finally, at the top, there are users or nodes that connect to the blockchain and perform various operations such as consensus, transaction verification, and processing. These concepts will be discussed in detail later in this book.
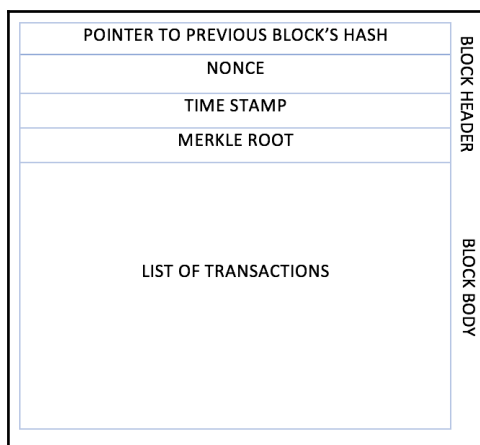
From a business standpoint, a blockchain can be defined as a platform where peers can exchange value / electronic cash using transactions without the need for a centrally-trusted arbitrator. For example, for cash transfers, banks act as a trusted third party. In financial trading, a central clearing house acts as an arbitrator between two trading parties. This concept is compelling, and once you absorb it, you will realize the enormous potential of blockchain technology. This disintermediation allows blockchain to be a decentralized consensus mechanism where no single authority is in charge of the database. Immediately, you'll see a significant benefit of decentralization here, because if no banks or central clearing houses are required, then it immediately leads to cost savings, faster transaction speeds, and trust.

A **block** is merely a selection of transactions bundled together and organized logically. A **transaction** is a record of an event, for example, the event of transferring cash from a sender's account to a beneficiary's account. A block is made up of transactions, and its size varies depending on the type and design of the blockchain in use.

A reference to a previous block is also included in the block unless it is a genesis block. A **genesis block** is the first block in the blockchain that is hardcoded at the time the blockchain was first started. The structure of a block is also dependent on the type and design of a blockchain. Generally, however, there are just a few attributes that are essential to the functionality of a block: the block header, which is composed of pointer to previous block, the timestamp, nonce, Merkle root, and the block body that contains transactions. There are also other attributes in a block, but generally, the aforementioned components are always available in a block.

A **nonce** is a number that is generated and used only once. A nonce is used extensively in many cryptographic operations to provide replay protection, authentication, and encryption. In blockchain, it's used in PoW consensus algorithms and for transaction replay protection.

Merkle root is a hash of all of the nodes of a Merkle tree. Merkle trees are widely used to validate the large data structures securely and efficiently. In the blockchain world, Merkle trees are commonly used to allow efficient verification of transactions. Merkle root in a blockchain is present in the block header section of a block, which is the hash of all transactions in a block. This means that verifying only the Merkle root is required to verify all transactions present in the Merkle tree instead of verifying all transactions one by one. We will elaborate further on these concepts in `Chapter 4`, *Public Key Cryptography*.
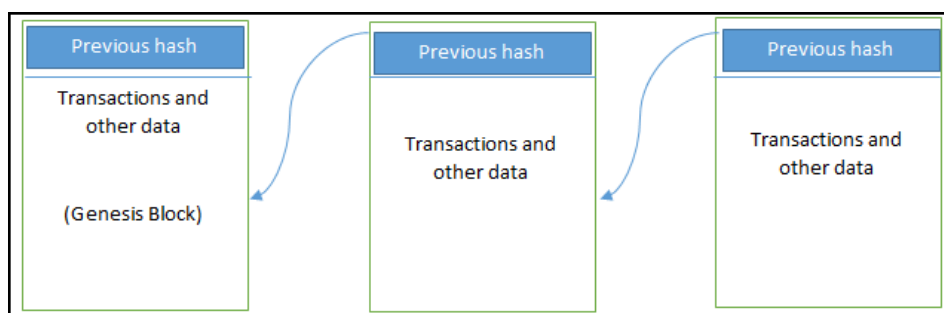
The generic structure of a block.

This preceding structure is a simple block diagram that depicts a block. Specific block structures relative to their blockchain technologies will be discussed later in the book with greater in-depth technical detail.

Generic elements of a blockchain

Now, let's walk through the generic elements of a blockchain. You can use this as a handy reference section if you ever need a reminder about the different parts of a blockchain. More precise elements will be discussed in the context of their respective blockchains in later chapters, for example, the Ethereum blockchain. The structure of a generic blockchain can be visualized with the help of the following diagram:

Generic structure of a blockchain

Elements of a generic blockchain are described here one by one. These are the elements that you will come across in relation to blockchain:

- **Address**: Addresses are unique identifiers used in a blockchain transaction to denote senders and recipients. An address is usually a public key or derived from a public key. While addresses can be reused by the same user, addresses themselves are unique. In practice, however, a single user may not use the same address again and generate a new one for each transaction. This newly-created address will be unique. Bitcoin is, in fact, a pseudonymous system. End users are usually not directly identifiable, but some research in removing the anonymity of Bitcoin users has shown that they can be identified successfully. A good practice is for users to generate a new address for each transaction in order to avoid linking transactions to the common owner, thus preventing identification.

- **Transaction**: A transaction is the fundamental unit of a blockchain. A transaction represents a transfer of value from one address to another.

- **Block**: A block is composed of multiple transactions and other elements, such as the previous block hash (hash pointer), timestamp, and nonce.

- **Peer-to-peer network**: As the name implies, a peer-to-peer network is a network topology wherein all peers can communicate with each other and send and receive messages.

- **Scripting or programming language**: Scripts or programs perform various operations on a transaction in order to facilitate various functions. For example, in Bitcoin, transaction scripts are predefined in a language called **Script**, which consist of sets of commands that allow nodes to transfer tokens from one address to another. Script is a limited language, however, in the sense that it only allows essential operations that are necessary for executing transactions, but it does not allow for arbitrary program development. Think of it as a calculator that only supports standard preprogrammed arithmetic operations. As such, Bitcoin script language cannot be called *Turing complete*. In simple words, Turing complete language means that it can perform any computation. It is named after Alan Turing who developed the idea of Turing machine that can run any algorithm however complex. Turing complete languages need loops and branching capability to perform complex computations. Therefore, Bitcoin's scripting language is not Turing complete, whereas Ethereum's Solidity language is.

To facilitate arbitrary program development on a blockchain, Turing complete programming language is needed, and it is now a very desirable feature of blockchains. Think of this as a computer that allows development of any program using programming languages. Nevertheless, the security of such languages is a crucial question and an essential and ongoing research area. We will discuss this in greater detail in `Chapter 5`, *Introducing Bitcoin*, `Chapter 9`, *Smart Contracts*, and `Chapter 13`, *Development Tools and Frameworks*, later in this book.

- **Virtual machine**: This is an extension of the transaction script introduced earlier. A *virtual machine* allows Turing complete code to be run on a blockchain (as smart contracts); whereas a transaction script is limited in its operation. However, virtual machines are not available on all blockchains. Various blockchains use virtual machines to run programs such as **Ethereum Virtual Machine** (**EVM**) and **Chain Virtual Machine** (**CVM**). EVM is used in Ethereum blockchain, while CVM is a virtual machine developed for and used in an enterprise-grade blockchain called **Chain Core**.
- **State machine**: A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next one and eventually to a final form by nodes on the blockchain network as a result of a transaction execution, validation, and finalization process.
- **Node**: A node in a blockchain network performs various functions depending on the role that it takes on. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This goal is achieved by following a **consensus protocol** (most commonly PoW). Nodes can also perform other functions such as simple payment verification (lightweight nodes), validation, and many other functions depending on the type of the blockchain used and the role assigned to the node. Nodes also perform a transaction signing function. Transactions are first created by nodes and then also digitally signed by nodes using private keys as proof that they are the legitimate owner of the asset that they wish to transfer to someone else on the blockchain network. This asset is usually a token or virtual currency, such as Bitcoin, but it can also be any real-world asset represented on the blockchain by using tokens.

- **Smart contract**: These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met. These programs are enforceable and automatically executable. The smart contract feature is not available on all blockchain platforms, but it is now becoming a very desirable feature due to the flexibility and power that it provides to the blockchain applications. Smart contracts have many use cases, including but not limited to identity management, capital markets, trade finance, record management, insurance, and e-governance. Smart contracts will be discussed in more detail in `Chapter 9`, *Smart Contracts*.

How blockchain works

We have now defined and described blockchain. Now let's see how a blockchain actually works. Nodes are either *miners* who create new blocks and mint cryptocurrency (coins) or *block signers* who validates and digitally sign the transactions. A critical decision that every blockchain network has to make is to figure out that which node will append the next block to the blockchain. This decision is made using a *consensus mechanism*. The consensus mechanism will be described later in this chapter.

Now we will look at the how a blockchain validates transactions and creates and adds blocks to grow the blockchain.

How blockchain accumulates blocks

Now we will look at a general scheme for creating blocks. This scheme is presented here to give you a general idea of how blocks are generated and what the relationship is between transactions and blocks:

1. A node starts a transaction by first creating and then digitally signing it with its private key. A transaction can represent various actions in a blockchain. Most commonly this is a data structure that represents transfer of value between users on the blockchain network. Transaction data structure usually consists of some logic of transfer of value, relevant rules, source and destination addresses, and other validation information. This will be covered in more detail in specific chapters on Bitcoin and Ethereum later in the book.
2. A transaction is propagated (flooded) by using a flooding protocol, called Gossip protocol, to peers that validate the transaction based on preset criteria. Usually, more than one node are required to verify the transaction.

3. Once the transaction is validated, it is included in a block, which is then propagated onto the network. At this point, the transaction is considered confirmed.

4. The newly-created block now becomes part of the ledger, and the next block links itself cryptographically back to this block. This link is a hash pointer. At this stage, the transaction gets its second confirmation and the block gets its first confirmation.

5. Transactions are then reconfirmed every time a new block is created. Usually, six confirmations in the Bitcoin network are required to consider the transaction final.

It is worth noting that steps 4 and 5 are considered non-compulsory, as the transaction itself is finalized in step 3; however, block confirmation and further transaction reconfirmations, if required, are then carried out in step 4 and step 5.

This completes the basic introduction to blockchain. In the next section, you will learn about the benefits and limitations of this technology.

Benefits and limitations of blockchain

Numerous advantages of blockchain technology have been discussed in many industries and proposed by thought leaders around the world who are participating in the blockchain space. The notable benefits of blockchain technology are as follows:

- **Decentralization**: This is a core concept and benefit of the blockchain. There is no need for a trusted third party or intermediary to validate transactions; instead, a consensus mechanism is used to agree on the validity of transactions.

- **Transparency and trust**: Because blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent. As a result, trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion in relation to selecting beneficiaries needs to be restricted.

- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not genuinely immutable, but because changing data is so challenging and nearly impossible, this is seen as a benefit to maintaining an immutable ledger of transactions.

- **High availability**: As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on every node, the system becomes highly available. Even if some nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available. This redundancy results in high availability.
- **Highly secure**: All transactions on a blockchain are cryptographically secured and thus provide network integrity.
- **Simplification of current paradigms**: The current blockchain model in many industries, such as finance or health, is somewhat disorganized. In this model, multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. However, as a blockchain can serve as a single shared ledger among many interested parties, this can result in simplifying the model by reducing the complexity of managing the separate systems maintained by each entity.
- **Faster dealings**: In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by enabling the quick settlement of trades. Blockchain does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed-upon data is already available on a shared ledger between financial organizations.
- **Cost saving**: As no trusted third party or clearing house is required in the blockchain model, this can massively eliminate overhead costs in the form of the fees which are paid to such parties.

As with any technology, some challenges need to be addressed in order to make a system more robust, useful, and accessible. Blockchain technology is no exception. In fact, much effort is being made in both academia and industry to overcome the challenges posed by blockchain technology. The most sensitive blockchain problems are as follows:

- Scalability
- Adaptability
- Regulation
- Relatively immature technology
- Privacy

All of these issues and possible solutions will be discussed in detail in `Chapter 18`, *Scalability and Other Challenges*.

Tiers of blockchain technology

In this section, various layers of blockchain technology are presented. It is thought that due to the rapid development and progress being made in blockchain technology, many applications will evolve. Some of these advancements have already been realized, while others are anticipated in the near future based on the current rate of advancement in blockchain technology.

The three levels discussed here were initially described in the book *Blockchain: Blueprint for a New Economy* by *Melanie Swan, O'Reilly Media, 2015* as blockchain tiers categorized by applications in each category. This is how blockchain is evolving, and this versioning shows different tiers of evolution and usage of blockchain technology. In fact, all blockchain platforms, with limited exceptions, support these functionalities and applications. This versioning is just a logical segregation of various blockchain categories based on the way that they are currently being used, are evolving, or predicted to evolve.

Also note that this versioning is being presented here for completeness and for historic reasons, as these definitions are somewhat blurred now, and with the exception of Bitcoin (Blockchain 1.0), all newer blockchain platforms that support smart contract development can be programmed to provide the functionalities and applications mentioned in all blockchain tiers: 1.0, 2.0, 3.0, and beyond.

In addition to Tier 1, Tier 2 and Tier 3, or Tier X in the future, the following represents my own vision of what blockchain technology eventually could become as this technology advances:

- **Blockchain 1.0**: This tier was introduced with the invention of Bitcoin, and it is primarily used for cryptocurrencies. Also, as Bitcoin was the first implementation of cryptocurrencies, it makes sense to categorize this first generation of blockchain technology to include only cryptographic currencies. All alternative cryptocurrencies as well as Bitcoin fall into this category. It includes core applications such as payments and applications. This generation started in 2009 when Bitcoin was released and ended in early 2010.
- **Blockchain 2.0**: This second blockchain generation is used by financial services and smart contracts. This tier includes various financial assets, such as derivatives, options, swaps, and bonds. Applications that go beyond currency, finance, and markets are incorporated at this tier. Ethereum, Hyperledger, and other newer blockchain platforms are considered part of Blockchain 2.0. This generation started when ideas related to using blockchain for other purposes started to emerge in 2010.

- **Blockchain 3.0**: This third blockchain generation is used to implement applications beyond the financial services industry and is used in government, health, media, the arts, and justice. Again, as in Blockchain 2.0, Ethereum, Hyperledger, and newer blockchains with the ability to code smart contracts are considered part of this blockchain technology tier. This generation of blockchain emerged around 2012 when multiple applications of blockchain technology in different industries were researched.

- **Blockchain X.0**: This generation represents a vision of blockchain singularity where one day there will be a public blockchain service available that anyone can use just like the Google search engine. It will provide services for all realms of society. It will be a public and open distributed ledger with general-purpose rational agents (*Machina economicus*) running on a blockchain, making decisions, and interacting with other intelligent autonomous agents on behalf of people, and regulated by code instead of law or paper contracts. This does not mean that law and contracts will disappear, instead law and contracts will be implementable in code.

Machina Economicus is a concept which comes from the field of **Artificial Intelligence** (**AI**) and computational economics. It can be defined as a machine that makes logical and perfect decisions. There are various technical challenges that need to be addressed before this dream can be realized.

> Discussion of Machina Economicus is beyond the scope of this book, interested readers can refer to `https://www.infosys.com/insights/purposeful-ai/Documents/machina-economicus.pdf`, for more information.

This concept in the context of blockchain and its convergence with AI will be elaborated on in `Chapter 19`, *Current Landscape and What's Next*.

Features of a blockchain

A blockchain performs various functions which are supported by various features. These functions include but are not limited to transfer of value, managing assets and agreements. All of the blockchain tiers described in the previous section perform these functions with the help of features offered by blockchain, but with some exceptions. For example, smart contracts are not supported by all blockchain platforms, such as Bitcoin. Another example is that not all blockchain platforms produce cryptocurrency or tokens, such as Hyperledger Fabric, and MultiChain.

The features of a blockchain are described here:

- **Distributed consensus**: Distributed consensus is the primary underpinning of a blockchain. This mechanism allows a blockchain to present a single version of the truth, which is agreed upon by all parties without the requirement of a central authority.
- **Transaction verification**: Any transactions posted from the nodes on the blockchain are verified based on a predetermined set of rules. Only valid transactions are selected for inclusion in a block.
- **Platform for smart contracts**: A blockchain is a platform on which programs can run to execute business logic on behalf of the users. Not all blockchains have a mechanism to execute *smart contracts*; however, this is a very desirable feature, and it is available on newer blockchain platforms such as Ethereum and MultiChain.

> **Smart Contracts**
>
> Blockchain technology provides a platform for running smart contracts. These are automated, autonomous programs that reside on the blockchain network and encapsulate the business logic and code needed to execute a required function when certain conditions are met. For example, think about an insurance contract where a claim is paid to the traveler if the flight is canceled. In the real world, this process normally takes a significant amount of time to make the claim, verify it, and pay the insurance amount to the claimant (traveler). What if this whole process were automated with cryptographically-enforced trust, transparency, and execution so that as soon as the smart contract received a feed that the flight in question has been canceled, it automatically triggers the insurance payment to the claimant? If the flight is on time, the smart contract pays itself.
>
> This is indeed a revolutionary feature of blockchain, as it provides flexibility, speed, security, and automation for real-world scenarios that can lead to a completely trustworthy system with significant cost reductions. Smart contracts can be programmed to perform any actions that blockchain users need and according to their specific business requirements.

- **Transferring value between peers**: Blockchain enables the transfer of value between its users via tokens. Tokens can be thought of as a carrier of value.

- **Generation of cryptocurrency**: This feature is optional depending on the type of blockchain in use. A blockchain can create cryptocurrency as an incentive to its miners who validate the transactions and spend resources to secure the blockchain. We will discuss cryptocurrencies in great detail in `Chapter 5`, *Introducing Bitcoin*.

- **Smart property**: It is now possible to link a digital or physical asset to the blockchain in such a secure and precise manner that it cannot be claimed by anyone else. You are in full control of your asset, and it cannot be double-spent or double-owned. Compare this with a digital music file, for example, which can be copied many times without any controls. While it is true that many **Digital Rights Management** (**DRM**) schemes are being used currently along with copyright laws, but none of them is enforceable in such a way as blockchain based DRM can be. Blockchain can provide DRM functionality in such a way that it can be enforced fully. There are famously broken DRM schemes which looked great in theory but were hacked due to one limitation or another. One example is Oculus hack (`http://www.wired.co.uk/article/oculus-rift-drm-hacked`).

Another example is PS3 hack, also copyrighted digital music, films and e-books are routinely shared on the internet without any limitations. We have copyright protection in place for many years, but digital piracy refutes all attempts to fully enforce the law on a blockchain, however, if you own an asset, no one else can claim it unless you decide to transfer it. This feature has far-reaching implications, especially in DRM and electronic cash systems where double-spend detection is a crucial requirement. The double-spend problem was first solved without the requirement of a trusted third party in Bitcoin.

- **Provider of security**: The blockchain is based on proven cryptographic technology that ensures the integrity and availability of data. Generally, confidentiality is not provided due to the requirements of transparency. This limitation is the leading barrier to its adoption by financial institutions and other industries that require privacy and confidentiality of transactions. As such, the privacy and confidentiality of transactions on the blockchain is being researched very actively, and advancements are already being made. It could be argued that, in many situations, confidentiality is not needed and transparency is preferred. For example, with Bitcoin, confidentiality is not an absolute requirement; however, it is desirable in some scenarios. A more recent example is Zcash, which provides a platform for conducting anonymous transactions. This scheme will be discussed in detail in `Chapter 8`, *Alternative Coins*. Other security services, such as non-repudiation and authentication, are also provided by blockchain, as all

actions are secured using private keys and digital signatures.

- **Immutability**: This is another critical feature of blockchain: once records are added to the blockchain, they are immutable. There is the remote possibility of rolling back changes, but this is to be avoided at all costs as doing so would consume an exorbitant amount of computing resources. For example, with Bitcoin if a malicious user wants to alter previous blocks, then it would require computing the PoW once again for all those blocks that have already been added to the blockchain. This difficulty makes the records on a blockchain essentially immutable.

- **Uniqueness**: This blockchain feature ensures that every transaction is unique and has not already been spent (double-spend problem). This feature is especially relevant with cryptocurrencies, where detection and avoidance of double spending are a vital requirement.

Types of blockchain

Based on the way that blockchain has evolved over the last few years, it can be divided into multiple categories with distinct though sometimes partially-overlapping attributes. You *should* note that the tiers described earlier in the chapter are a different concept whereby the logical categorization of blockchain based on its evolution and usage is presented.

In this section, we will examine the different types of blockchains from a technical and business usage perspective. These blockchain types can occur on any blockchain tier, as there is no direct relationship between those tiers and the various types of blockchain.

In this section we'll examine:

- Distributed ledgers
- Distributed Ledger Technology (DLT)
- Blockchains
- Ledgers

Distributed ledgers

First, I need to clarify an ambiguity. It should be noted that a *distributed ledger* is a broad term describing shared databases; hence, all blockchains technically fall under the umbrella of shared databases or distributed ledgers. Although all blockchains are fundamentally distributed ledgers, all distributed ledgers are not necessarily a blockchain.

A critical difference between a distributed ledger and blockchain is that a distributed ledger does not necessarily consist of blocks of transactions to keep the ledger growing. Rather, a blockchain is a special type of shared database that is comprised of blocks of transactions. An example of a distributed ledger that does not use blocks of transactions is R3's Corda. Corda is a distributed ledger which is developed to record and manage agreements and is especially focused on financial services industry. On the other hand, more widely-known blockchains like Bitcoin and Ethereum make use of blocks to update the shared database.

As the name suggests, a distributed ledger is distributed among its participants and spread across multiple sites or organizations. This type of ledger can be either private or public. The fundamental idea here is that, unlike many other blockchains, the records are stored contiguously instead of being sorted into blocks. This concept is used in Ripple which is a blockchain and cryptocurrency based global payment network.

Distributed Ledger Technology

It should be noted that over the last few years, the terms distributed ledger or **Distributed Ledger Technology** (**DLT**) have grown to be commonly used to describe blockchain in finance industry. Sometimes, blockchain and DLT are used interchangeably. Though this is not entirely accurate, it is how the term has evolved recently, especially in the finance sector. In fact, DLT is now a very active and thriving area of research in the financial sector. From a financial sector point of view, DLTs are permissioned blockchains that are shared and used between known participants. DLTs usually serve as a shared database, with all participants known and verified. They do not have a cryptocurrency or do not require mining to secure the ledger.

Public blockchains

As the name suggests, public blockchains are not owned by anyone. They are open to the public, and anyone can participate as a node in the decision-making process. Users may or may not be rewarded for their participation. All users of these *permissionless* or *unpermissioned* ledgers maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism to decide the eventual state of the ledger. Bitcoin and Ethereum are both considered public blockchains.

Private blockchains

As the name implies, private blockchains are just that—private. That is, they are open only to a consortium or group of individuals or organizations who have decided to share the ledger among themselves. There are various blockchains now available in this category, such as HydraChain and Quorum. Optionally, both of these blockchains can also run in public mode if required, but their primary purpose is to provide a private blockchain.

Semiprivate blockchains

With *semiprivate blockchains*, part of the blockchain is private and part of it is public. Note that this is still just a concept today, and no real world POCs have yet been developed. With a semi-private blockchain, the private part is controlled by a group of individuals, while the public part is open for participation by anyone.

This hybrid model can be used in scenarios where the private part of the blockchain remains internal and shared among known participants, while the public part of the blockchain can still be used by anyone, optionally allowing mining to secure the blockchain. This way, the blockchain as a whole can be secured using PoW, thus providing consistency and validity for both the private and public parts. This type of blockchain can also be called a *semi-decentralized* model, where it is controlled by a single entity but still allows for multiple users to join the network by following appropriate procedures.

Sidechains

More precisely known as *pegged sidechains*, this is a concept whereby coins can be moved from one blockchain to another and moved back again. Typical uses include the creation of new *altcoins* (alternative cryptocurrencies) whereby coins are burnt as a proof of an adequate stake. *Burnt* or *burning the coins* in this context means that the coins are sent to an address which is unspendable and this process makes the *burnt* coins irrecoverable. This mechanism is used to bootstrap a new currency or introduce scarcity which results in increased value of the coin.

This mechanism is also called **Proof of Burn** (**PoB**) and is used as an alternative method for distributed consensus to PoW and **Proof of Stake** (**PoS**). The aforementioned example for burning coins applies to a **one-way pegged sidechain**. The second type is called a **two-way pegged sidechain**, which allows the movement of coins from the main chain to the sidechain and back to the main chain when required.

This process enables the building of smart contracts for the Bitcoin network. Rootstock is one of the leading examples of a sidechain, which enables smart contract development for Bitcoin using this paradigm. It works by allowing a two-way peg for the Bitcoin blockchain, and this results in much faster throughput.

Permissioned ledger

A *permissioned ledger* is a blockchain where participants of the network are already known and trusted. Permissioned ledgers do not need to use a distributed consensus mechanism; instead, an agreement protocol is used to maintain a shared version of the truth about the state of the records on the blockchain. In this case, for verification of transactions on the chain, all verifiers are already preselected by a central authority and typically there is no need for a mining mechanism.

By definition, there is also no requirement for a permissioned blockchain to be private, as it can be a public blockchain but with regulated access control. For example, Bitcoin can become a permissioned ledger if an access control layer is introduced on top of it that verifies the identity of a user and then allows access to the blockchain.

Shared ledger

This is a generic term that is used to describe any application or database that is shared by the public or a consortium. Generally, all blockchains, fall into the category of a shared ledger.

Fully private and proprietary blockchains

There is no mainstream application of these types of blockchains, as they deviate from the core concept of decentralization in blockchain technology. Nonetheless, in specific private settings within an organization, there could be a need to share data and provide some level of guarantee of the authenticity of the data.

An example of this type of blockchain might be to allow for collaboration and the sharing data between various government departments. In that case, no complex consensus mechanism is required, apart from simple state machine replication and an agreement protocol with known central validators. Even in private blockchains, tokens are not really required, but they can be used as means of transferring value or representing some real-world asset.

Tokenized blockchains

These blockchains are standard blockchains that generate cryptocurrency as a result of a consensus process via mining or initial distribution. Bitcoin and Ethereum are prime examples of this type of blockchain.

Tokenless blockchains

These blockchains are designed in such a way that they do not have the basic unit for the transfer of value. However, they are still valuable in situations where there is no need to transfer value between nodes and only the sharing of data among various trusted parties is required. This is similar to full private blockchains, the only difference being that use of tokens is not required. This can also be thought of as a shared distributed ledger used for storing data. It does have its benefits when it comes to immutability, security, and consensus driven updates but are not used for common blockchain application of value transfer or cryptocurrency.

This ends our examination of the various type of blockchain, we'll now move in the next section to discuss the concept of census.

Consensus

Consensus is the backbone of a blockchain and, as a result, it provides decentralization of control through an optional process known as **mining**. The choice of the **consensus algorithm** is also governed by the type of blockchain in use; that is, not all consensus mechanisms are suitable for all types of blockchains. For example, in public permissionless blockchains, it would make sense to use PoW instead of a simple agreement mechanism that is perhaps based on proof of authority. Therefore, it is essential to choose an appropriate consensus algorithm for a particular blockchain project.

Consensus is a process of agreement between distrusting nodes on the final state of data. To achieve consensus, different algorithms are used. It is easy to reach an agreement between two nodes (in client-server systems, for example), but when multiple nodes are participating in a distributed system and they need to agree on a single value, it becomes quite a challenge to achieve consensus. This process of attaining agreement common state or value among multiple nodes despite the failure of some nodes is known as **distributed consensus**.

Consensus mechanism

A **consensus mechanism** is a set of steps that are taken by most or all nodes in a blockchain to agree on a proposed state or value. For more than three decades, this concept has been researched by computer scientists in industry and academia. Consensus mechanisms have most recently come into the limelight and gained considerable popularity with the advent of blockchain and Bitcoin.

There are various requirements that must be met to provide the desired results in a consensus mechanism. The following describes these requirements:

- **Agreement**: All honest nodes decide on the same value
- **Termination**: All honest nodes terminate execution of the consensus process and eventually reach a decision
- **Validity**: The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node
- **Fault tolerant**: The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes)
- **Integrity**: This is a requirement that no node can make the decision more than once in a single consensus cycle

Types of consensus mechanisms

All consensus mechanisms are developed to deal with faults in a distributed system and to allow distributed systems to reach a final state of agreement. There are two general categories of consensus mechanisms. These categories deal with all types of faults (fail stop type or arbitrary). These common types of consensus mechanisms are as follows:

- **Traditional Byzantine Fault Tolerance (BFT)-based**: With no compute-intensive operations, such as partial hash inversion (as in Bitcoin PoW), this method relies on a simple scheme of nodes that are publisher-signed messages. Eventually, when a certain number of messages are received, then an agreement is reached.
- **Leader election-based consensus mechanisms**: This arrangement requires nodes to compete in a leader-election lottery, and the node that wins proposes a final value. For example, the PoW used in Bitcoin falls into this category.

Many practical implementations of consensus protocols have been proposed. **Paxos** is the most famous of these protocols. It was introduced by Leslie Lamport in 1989. With Paxos, nodes are assigned various roles such as Proposer, Acceptor, and Learner. Nodes or processes are named replicas, and consensus is achieved in the presence of faulty nodes by agreement among a majority of nodes.

An alternative to Paxos is RAFT, which works by assigning any of three states; that is, follower, candidate, or leader to the nodes. A leader is elected after a candidate node receives enough votes, and all changes then have to go through the leader. The leader commits the proposed changes once replication on the majority of the follower nodes is completed. More detail on the theory of consensus mechanisms from a distributed system point of view are beyond the scope of this chapter. Later in this chapter, however, a full section is dedicated to the introduction of consensus protocols. Specific algorithms will be discussed in chapters dedicated to Bitcoin and other blockchains later in this book.

Consensus in blockchain

Consensus is a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of the truth by all peers on the blockchain network. This concept was previously discussed in the distributed systems section of this chapter. In this section, we will address consensus in the context of blockchain technology. Some concepts presented here are still relevant to distributed systems theory, but they are explained from a blockchain perspective.

Roughly, the following describes the two main categories of consensus mechanisms:

- Proof-based, leader-election lottery based, or the Nakamoto consensus whereby a leader is elected at random (using an algorithm) and proposes a final value. This category is also referred to as the *fully decentralized* or *permissionless* type of consensus mechanism. This type is well used in the Bitcoin and Ethereum blockchain in the form of a PoW mechanism.
- BFT-based is a more traditional approach based on rounds of votes. This class of consensus is also known as the *consortium* or *permissioned* type of consensus mechanism.

BFT-based consensus mechanisms perform well when there are a limited number of nodes, but they do not scale well. On the other hand, leader-election lottery based (PoW) type consensus mechanisms scale very well but perform very slowly. As there is significant research being conducted in this area, new types of consensus mechanism are also emerging, such as the semi-decentralized type, which is used in the Ripple network. Ripple network will be discussed in detail in `Chapter 16`, *Alternative Blockchains*. There are also various other proposals out there, which are trying to find the right balance between scalability and performance. Some notable projects include PBFT, Hybrid BFT, BlockDAG, Tezos, Stellar, and GHOST.

The consensus algorithms available today, or that are being researched in the context of blockchain, are presented here. The following is not an exhaustive list, but it includes all notable algorithms.

- **Proof of Work (PoW)**: This type of consensus mechanism relies on proof that adequate computational resources have been spent before proposing a value for acceptance by the network. This scheme is used in Bitcoin, Litecoin, and other cryptocurrency blockchains. Currently, it is the only algorithm that has proven to be astonishingly successful against any collusion attacks on a blockchain network, such as the Sybil attack. The Sybil attack will be discussed in `Chapter 5`, *Introducing Bitcoin*.
- **Proof of Stake (PoS)**: This algorithm works on the idea that a node or user has an adequate stake in the system; that is, the user has invested enough in the system so that any malicious attempt by that user would outweigh the benefits of performing such an attack on the network. This idea was first introduced by Peercoin, and it is going to be used in the Ethereum blockchain version called *Serenity*. Another important concept in PoS is **coin age**, which is a criterion derived from the amount of time and number of coins that have not been spent. In this model, the chances of proposing and signing the next block increase with the coin age.

- **Delegated Proof of Stake (DPoS)**: This is an innovation over standard PoS, whereby each node that has a stake in the system can delegate the validation of a transaction to other nodes by voting. It is used in the BitShares blockchain.

- **Proof of Elapsed Time (PoET)**: Introduced by Intel in 2016, PoET uses a **Trusted Execution Environment** (**TEE**) to provide randomness and safety in the leader election process via a guaranteed wait time. It requires the Intel **Software Guard Extensions** (**SGX**) processor to provide the security guarantee for it to be secure. This concept is discussed in more detail in `Chapter 15`, *Hyperledger*, in the context of the Intel's *Sawtooth Lake* blockchain project.

- **Proof of Deposit (PoD)**: In this case, nodes that wish to participate in the network have to make a security deposit before they can mine and propose blocks. This mechanism is used in the Tendermint blockchain.

- **Proof of Importance (PoI)**: This idea is significant and different from PoS. PoI not only relies on how large a stake a user has in the system, but it also monitors the usage and movement of tokens by the user in order to establish a level of trust and importance. It is used in the NEM coin blockchain. More information about this coin is available at NEM's website `https://nem.io`.

- **Federated consensus or federated Byzantine consensus**: This mechanism is used in the stellar consensus protocol. Nodes in this protocol retain a group of publicly-trusted peers and propagate only those transactions that have been validated by the majority of trusted nodes.

- **Reputation-based mechanisms**: As the name suggests, a leader is elected by the reputation it has built over time on the network. It is based on the votes of other members.

- **PBFT**: This mechanism achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols including PBFT, PAXOS, RAFT, and **Federated Byzantine Agreement** (**FBA**) are also being used or have been proposed for use in many different implementations of distributed systems and blockchains.

- **Proof of Activity** (**PoA**): This scheme is a combination of PoS and PoW, which ensures that a stakeholder is selected in a pseudorandom but uniform fashion. This is a comparatively more energy-efficient mechanism as compared to PoW. It utilizes a new concept called *Follow the Satoshi*. In this scheme, PoW and PoS are combined together to achieve consensus and good level of security. This scheme is more energy efficient as PoW is used only in the first stage of the mechanism, after the first stage it switches to PoS which consumes negligible energy. We will discuss these ideas further in `Chapter 6`, *Bitcoin Network and Payments* where protocols are reviewed in the context of advanced Bitcoin protocols.

- **Proof of Capacity (PoC)**: This scheme uses hard disk space as a resource to mine the blocks. This is different from PoW, where CPU resources are used. In in PoC, hard disk space is utilized for mining and as such is also known as *hard drive mining*. This concept was first introduced in the Burstcoin cryptocurrency.
- **Proof of Storage (PoS)**: This scheme allows for the outsourcing of storage capacity. This scheme is based on the concept that a particular piece of data is probably stored by a node *which* serves as a means to participate in the consensus mechanism. Several variations of this scheme have been proposed, such as Proof of Replication, Proof of Data Possession, Proof of Space, and Proof of Space-Time.

CAP theorem and blockchain

CAP theorem, also known as Brewer's theorem, was introduced by Eric Brewer in 1998 as conjecture. In 2002, it was proven as a theorem by Seth Gilbert and Nancy Lynch. The theory states that any distributed system cannot have consistency, availability, and partition tolerance simultaneously:

- **Consistency** is a property which ensures that all nodes in a distributed system have a single, current, and identical copy of the data.
- **Availability** means that the nodes in the system are up, accessible for use, and are accepting incoming requests and responding with data without any failures as and when required. In other words, data is available at each node and the nodes are responding to requests.
- **Partition tolerance** ensures that if a group of nodes is unable to communicate with other nodes due to network failures, the distributed system continues to operate correctly. This can occur due to network and node failures.

It has been proven that a distributed system cannot have consistency, availability, and partition tolerance simultaneously. This is explained with the following example. Let's imagine that there is a distributed system with two nodes. Now let us apply the three theorem properties on this smallest of possible distributed systems only with two nodes.

- **Consistency** is achieved if both nodes have the same shared state; that is, they have the same up-to-date copy of the data.
- **Availability** is achieved if both nodes are up and running and responding with the latest copy of data.
- **Partition tolerance** is achieved if communication does not break down between two nodes (either due to network issues, Byzantine faults, and so forth), and they are able to communicate with each other.

Now think of scenario where a partition occurs and nodes can no longer communicate with each other. If no new updated data comes in, it can only be updated on one node only. In that case, if the node accepts the update, then only that one node in the network is updated and therefore consistency is lost. Now, if the update is rejected by the node, that would result in loss of availability. In that case due to partition tolerance, both availability and consistency are unachievable.

This is strange because somehow blockchain manages to achieve all of these properties—or does it? This will be explained shortly. To achieve fault tolerance, replication is used. This is a standard and widely-used method to achieve fault tolerance. Consistency is achieved using consensus algorithms in order to ensure that all nodes have the same copy of the data. This is also called **state machine replication**. The blockchain is a means for achieving state machine replication. In general, there are two types of faults that a node can experience. Both of these types fall under the broader category of faults that can occur in a distributed system:

- **Fail-stop fault**: This type of fault occurs when a node merely has crashed. Fail-stop faults are the easier ones to deal with of the two fault types. Paxos protocol, introduced earlier in this chapter, is normally used to deal with this type of fault. These faults are simple to deal with

- **Byzantine faults**: The second type of fault is one where the faulty node exhibits malicious or inconsistent behavior arbitrarily. This type is difficult to handle since it can create confusion due to misleading information. This can be a result of an attack by adversaries, a software bug, or data corruption. State machine replication protocols such as PBFT was developed to address this second type of faults.

Strangely, it seems that the CAP theorem is violated in the blockchain, especially in its most successful implementation, Bitcoin. However, this is not the case. In blockchains, consistency is sacrificed in favor of availability and partition tolerance. In this scenario, **Consistency** (**C**) on the blockchain is not achieved simultaneously with **Partition tolerance** (**P**) and **Availability** (**A**), but it is achieved over time. This is called eventual consistency, where consistency is achieved as a result of validation from multiple nodes over time. The concept of mining was introduced in Bitcoin for this purpose. **Mining** is a process that facilitates the achievement of consensus by using the PoW consensus algorithm. At a higher level, mining can be defined as a process that is used to add more blocks to the blockchain. More on this later in `Chapter 5`, *Introducing Bitcoin*.

Summary

In this chapter, we introduced you to the blockchain technology at an advanced level. First, we discussed some basic concepts of distributed systems, and then we reviewed the history of blockchain. We also discussed concepts such as electronic cash.

Furthermore, we presented various definitions of blockchain from different point of views. We also introduced you to few applications of blockchain technology. Next, we explored different types of blockchain. Finally, we examined the benefits and limitations of this new technology. Few topics such as blockchain scalability and adaptability issues were intentionally introduced only briefly, as these will be discussed in depth, in later chapters.

In the next chapter, we will introduce you to the concept of decentralization, which is central to the idea behind blockchains and their vast number of applications.