1
Blockchain 101

It is very likely that anyone reading this book has already heard about blockchain and has some basic appreciation of its enormous potential.

With the invention of bitcoin in 2008 the world was introduced to a new concept that is now likely to revolutionize the whole of society. It's something that has promised to impact every industry including but not limited to finance, government, and media. Some describe it as a revolution whereas another school of thought says that it's going to be an evolution and it will take many years before any practical benefits from blockchain come to fruition. This is correct to some extent but in my opinion the revolution has already started; many big 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 at the preliminary exploration stage but are expected to progress more quickly as the technology is now becoming more mature. It is a technology that has an impact on current technologies too and possesses the ability to change them at a fundamental level.

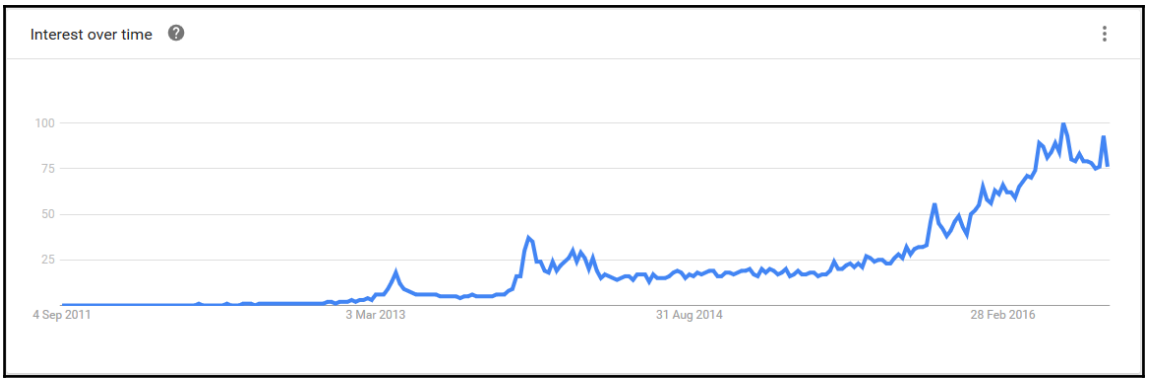
According to Gartner's technology hype cycle graph shown below, the blockchain technology is currently at the *peak of inflated expectations* (as of July 2016) and is expected to be ready for mainstream adoption in 5 to 10 years:

Figure 1. Hype Cycle for Emerging Technologies, 2016

Source: Gartner (August 2016)

Gartner's hype cycle for emerging technologies

Interest in blockchain technology has soared in the last few years and, once disregarded by some as geek money from a cryptocurrency point of view or as something that was not really considered worthwhile, it is now being researched by the largest companies and organizations around the world with millions of dollars being spent in order to adopt and experiment with this technology. A simple trend search on Google reveals the scale of interest in the blockchain technology over the last few years:



Google trends for blockchain

Various benefits of this technology are being envisaged such as decentralized trust, cost savings, transparency, and efficiency. However, there are various challenges too that are an area of active research such as scalability and privacy. `Chapter 12`, *Scalablity and Other Challenges* is dedicated to a discussion of the limitations and challenges of blockchain technology.

This chapter is an introduction to blockchain technology, its technical foundations, the theory behind it, and various technologies that have been combined together in order to build what is known today as blockchain.

In 2008 a groundbreaking paper *Bitcoin: A Peer-to-Peer Electronic Cash System* was written on the topic of peer-to-peer electronic cash under the pseudonym *Satoshi Nakamoto* and introduced the term *chain of blocks*. This term over the years has now evolved into the word blockchain.

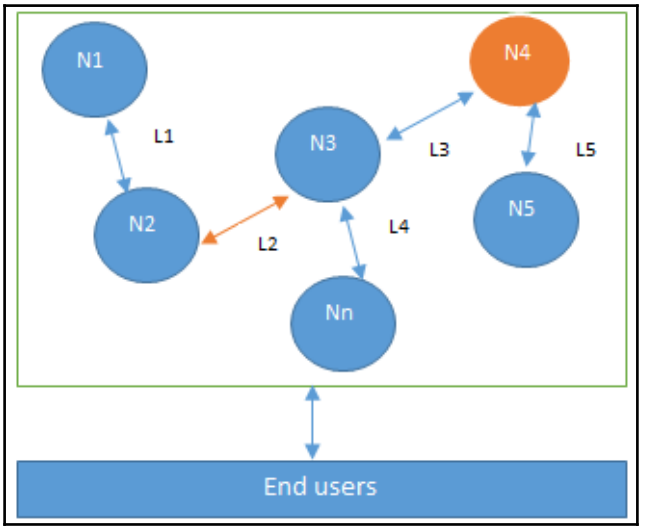
In this chapter, first the theoretical foundations of distributed systems are described, then the precursors of bitcoin (with which blockchain technology was introduced) such as e-cash and hashcash, and then finally the blockchain technology is introduced. This is a logical way of understanding blockchain technology because the roots of blockchain are in distributed systems.

Distributed systems

Understanding distributed systems is essential in order to understand blockchain because basically blockchain at its core is a distributed system. More precisely 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 in order to achieve a common outcome and it's modeled in such a way that end users see it as a single logical 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 have their own memory and processor. A node that can exhibit arbitrary behavior is also known as a Byzantine node. This arbitrary behavior can be intentionally malicious, which is detrimental to the operation of the network. Generally, any unexpected behavior of a node on the network can be categorized as Byzantine. This term arbitrarily encompasses any behavior that is unexpected or malicious:



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

The main 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 tolerate this and should continue to work flawlessly in order to achieve the desired result. This has been an area of active research for many years and several algorithms and mechanisms has been proposed to overcome these issues.

Distributed systems are so challenging to design that a theorem known as the CAP theorem has been proved and states that a distributed system cannot have all much desired properties simultaneously. In the next section, a basic introduction to the CAP theorem will be provided.

CAP theorem

This is also known as Brewer's theorem, introduced originally by *Eric Brewer* as a conjecture in 1998; in 2002 it was proved as a theorem by *Seth Gilbert* and *Nancy Lynch*.

The theorem states that any distributed system cannot have Consistency, Availability, and Partition tolerance simultaneously:

- **Consistency** is a property that ensures that all nodes in a distributed system have a single latest copy of data
- **Availability** means that the system is up, accessible for use, and is accepting incoming requests and responding with data without any failures as and when required
- **Partition tolerance** ensures that if a group of nodes fails the distributed system still continues to operate correctly

It has been proven that a distributed system cannot have all the afore mentioned three properties at the same time. This is strange because somehow blockchain manages to achieve all these properties, or does it really? This will be explained later in the chapter where the CAP theorem in the context of blockchain is discussed.

In order to achieve fault tolerance, replication is used. This is a common and widely used method to achieve fault tolerance. Consistency is achieved using consensus algorithms to ensure that all nodes have the same copy of data. This is also called **state machine replication**. Blockchain is basically a method to achieve state machine replication.

In general there are two types of fault that a node can experience: where a faulty node has simply crashed and where the faulty node can exhibit malicious or inconsistent behavior arbitrarily. This is the type which is difficult to deal with since it can cause confusion due to misleading information.

Byzantine Generals problem

Before discussing consensus in distributed systems, events in history are presented that are precursors to the development of successful and practical consensus mechanisms.

In September 1962, *Paul Baran* introduced the idea of cryptographic signatures with his paper *On distributed communications networks*. This is the paper where the concept of decentralized networks was also introduced for the very first time. Then in 1982 a thought experiment was proposed by *Lamport et al.* whereby a group of army generals who are leading different parts of the Byzantine army are planning to attack or retreat from a city. The only way of communication between them is a messenger and they need to agree to attack at the same time in order to win. The issue is that one or more generals can be traitors and can communicate a misleading message. Therefore there is a need to find a viable mechanism that allows agreement between generals even in the presence of treacherous generals so that the attack can still take place at the same time. As an analogy with distributed systems, generals can be considered as nodes, traitors can be considered Byzantine (malicious) nodes, and the messenger can be thought of as a channel of communication between the generals.

This problem was solved in 1999 by *Castro* and *Liskov* who presented the **Practical Byzantine Fault Tolerance** (**PBFT**) algorithm. Later on in 2009, the first practical implementation was made with the invention of bitcoin where the **Proof of Work** (**PoW**) algorithm was developed as a mechanism to achieve consensus.

Consensus

Consensus is a process of agreement between distrusting nodes on a final state of data. In order to achieve consensus different algorithms can be used. It is easy to reach an agreement between two nodes (for example in client-server systems) but when multiple nodes are participating in a distributed system and they need to agree on a single value it becomes very difficult to achieve consensus. This concept of achieving consensus between multiple nodes is known as distributed consensus.

Consensus mechanisms

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

There are various requirements which must be met in order to provide the desired results in a consensus mechanism. The following are their requirements with brief descriptions:

- **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 where by no node makes the decision more than once. The nodes make decisions only once in a single consensus cycle.

Types of consensus mechanism

There are various types of consensus mechanism; some common types are described as follows:

- **Byzantine fault tolerance-based**: With no compute intensive operations such as partial hash inversion, this method relies on a simple scheme of nodes that are publishing signed messages. Eventually, when a certain number of messages are received, then an agreement is reached.
- **Leader-based consensus mechanisms**: This type of mechanism requires nodes to compete for the *leader-election lottery* and the node that wins it proposes a final value.

Many practical implementations have been proposed such as **Paxos**, the most famous protocol introduced by **Leslie Lamport** in 1989. In 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.

Another 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 now have to go through the Leader, who commits the proposed changes once replication on the majority of follower nodes is completed.

More details about the theory of consensus mechanisms from a distributed system point of view is beyond the scope of this chapter. Later in this chapter, 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.

The history of blockchain

Blockchain was introduced with the invention of bitcoin in 2008 and then with its practical implementation in 2009. For this chapter, it is sufficient to introduce bitcoin very briefly as there is a full chapter on bitcoin later on but it is also essential to refer to bitcoin because without it, the history of blockchain is not complete.

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*.

Electronic cash

Just as understanding the concepts of distributed systems is necessary in order to understand blockchain technology, the idea of electronic cash is also essential to appreciate the first and astonishingly successful application of blockchain: the bitcoin, or broadly cryptocurrencies. Theoretical concepts in distributed systems such as consensus algorithms provided the basis of the practical implementation of Proof of Work algorithms in bitcoin; moreover, ideas from different electronic cash schemes also paved the way for the invention of cryptocurrencies, specifically bitcoin.

In this section, the reader will be introduced to the idea of electronic cash and then various other concepts that existed before cryptocurrencies that led to the development of bitcoin are presented.

The concept of electronic cash

Fundamental issues that need to be addressed in e-cash systems are accountability and anonymity. *David Chaum* addressed both of these issues in his seminal paper in 1984 by introducing two cryptographic operations, namely blind signatures and secret sharing. These terminologies and related concepts will be discussed in detail in `Chapter 3`, *Cryptography and Technical Foundations*. At the moment, it is sufficient to say that blind signatures allow signing a document without actually seeing it and secret sharing is a concept that allows the detection of using the same e-cash token twice (double spending).

After this other protocols emerged such as **Chaum, Fiat, and Naor** (**CFN**), e-cash schemes that introduced anonymity and double spending detection. Brand's e-cash is another system that improved on CFN, made it more efficient, and introduced the concept of security reduction to prove statements about the e-cash scheme. Security reduction is a technique used in cryptography to prove that a certain algorithm is secure by using another problem as a comparison. Put another way, a cryptographic security algorithm is as hard to break as some other hard problem; thus by comparison it can be deduced that the cryptographic security algorithm is secure too.

A different but relevant concept called **hashcash** was introduced by *Adam Back* in 1997 as a PoW system to control e-mail spam. The idea is quite simple: if legitimate users want to send e-mails then they are required to compute a hash as a proof that they have spent a reasonable amount of computing resources before sending the e-mail. Generating hashcash is a compute intensive process but does not inhibit a legitimate user from sending the e-mail because the usual number of e-mails required to be sent by a legitimate user is presumably quite low. On the other hand, if a spammer wants to send e-mails, usually thousands in number, then it becomes infeasible to compute hashcash for all e-mails, thus making the spamming effort expensive; as a result this mechanism can be used to thwart e-mail spamming. Hashcash takes a considerable amount of computing resources to compute but is easy and quick to verify. Verification is performed by the user who receives the e-mail. Hashcash is popularized by its use in the bitcoin mining process. This idea of using computational puzzles or pricing functions to prevent e-mail spam was introduced originally in 1992 by *Cynthia Dwork* and *Moni Naor*. Pricing function was the name given to the hard functions that are required to be computed before access to a resource can be granted. Later, *Adam Back* invented hashcash independently in 1997, which introduced the usage of computing hash functions as PoW.

In 1998 **b-money** was introduced by *Wei Dai* and proposed the idea of creating money via solving computational puzzles such as hashcash. It's based on a peer-to-peer network where each node maintains its own list of transactions.

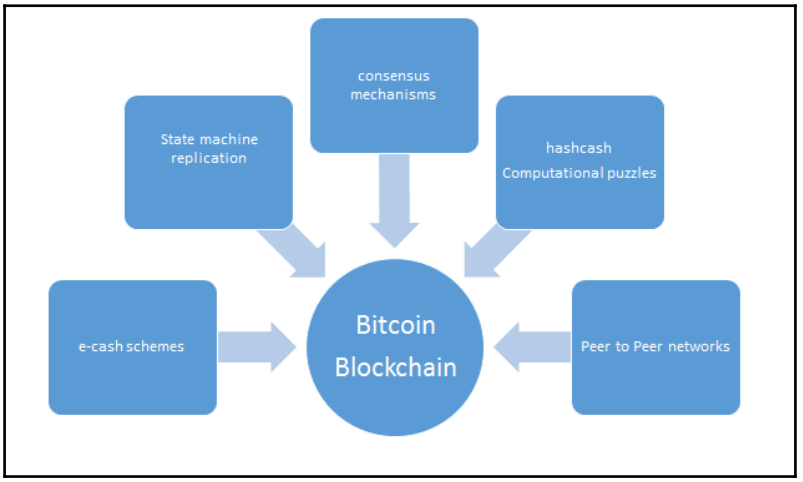
Another similar idea by *Nick Szabo* called BitGold was introduced in 2005 and also proposed solving computational puzzles to mint digital currency. In 2005 *Hal Finney* introduced the concept of cryptographic currency by combining ideas from b-money and hashcash puzzles but it still relied on a centralized trusted authority.

There were multiple issues with the schemes described in infeasible preceding paragraphs. These problems range from no clear solution of disagreements between nodes to reliance on a central trusted third party and trusted timestamping.

In 2009 the first practical implementation of a cryptocurrency named bitcoin was introduced; for the very first time it solved the problem of distributed consensus in a trustless network. It uses public key cryptography with hashcash as PoW to provide a secure, controlled, and decentralized method of minting digital currency. The key innovation is the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism. This will be explained in more detail in `Chapter 4`, *Bitcoin*.

Looking at all the aforementioned technologies and their history, it is easy to see how ideas and concepts from electronic cash schemes and distributed systems were combined together to invent bitcoin and what now is known as blockchain.

This can also be visualized with the help of the following diagram:



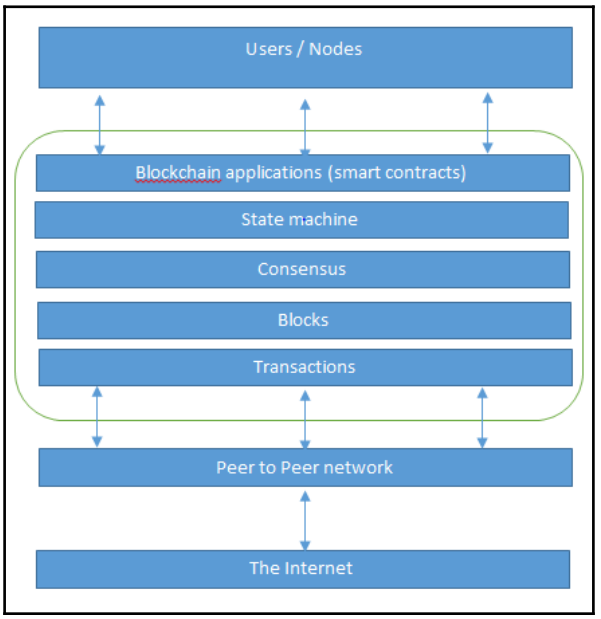
The various ideas that helped with the invention of bitcoin and blockchain

Introduction to blockchain

There are various definitions of blockchain; it depends on how you look at it. If you look at it from a business perspective it can be defined in that context, if you look at it from a technical perspective one can define it in view of that.

Blockchain at its core 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.

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 below in the diagram. It is analogous to SMTP, HTTP, or FTP running on top of TCP/IP. This is shown in the following diagram:

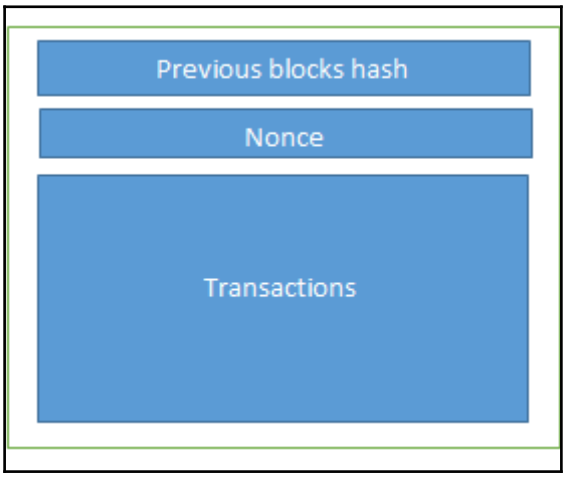


The network view of a blockchain

From a business point of view a blockchain can be defined as a platform whereby peers can exchange values using transactions without the need for a central trusted arbitrator. This is a powerful concept and once readers understand it they will realize the tsunamic potential of blockchain technology. This allows blockchain to be a decentralized consensus mechanism where no single authority is in charge of the database.

A block is simply a selection of transactions bundled together in order to organize them logically. It is made up of transactions and its size is variable 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's a genesis block. A genesis block is the first block in the blockchain that was hardcoded at the time the blockchain was started. The structure of a block is also dependent on the type and design of a blockchain, but generally there are a few attributes that are essential to the functionality of a block, such as the block header, pointers to previous blocks, the time stamp, nonce, transaction counter, transactions, and other attributes.

This is shown in a simple block diagram as follows. This is a general depiction of a block; specific block structures relative to their blockchain technologies will be discussed later in the book with more in-depth technical details:

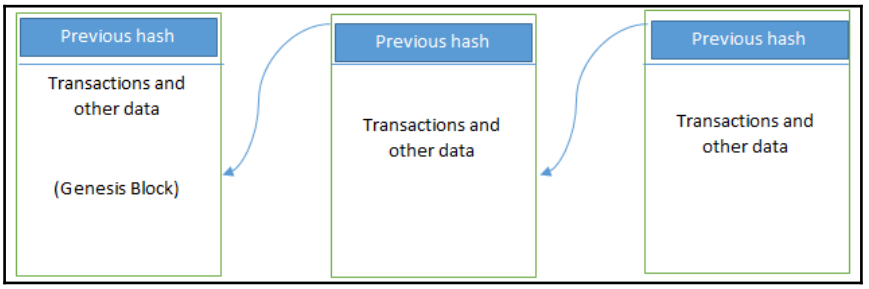


The structure of a block

Various technical definitions of blockchains

- Blockchain is a decentralized consensus mechanism. In a blockchain, all peers eventually come to an agreement regarding the state of a transaction.
- Blockchain is a distributed shared ledger. Blockchain can be considered a shared ledger of transactions. The transaction are ordered and grouped into blocks. Currently, the real-world model is based on private databases that each organization maintains whereas the distributed ledger can serve as a single source of truth for all member organizations that are using the blockchain.
- Blockchain is a data structure; it is basically a linked list that uses hash pointers instead of normal pointers. Hash pointers are used to point to the previous block.

The structure of a generic blockchain can be visualized with the help of the following diagram:



Generic structure of a blockchain

Generic elements of a blockchain

In this section, the generic elements of blockchain are presented. More precise elements will be discussed in the context of their respective blockchains in later chapters, for example, the Ethereum blockchain.

Addresses

Addresses are unique identifiers that are used in a transaction on the blockchain 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 generated address will be unique. Bitcoin is in fact a pseudonymous system. End users are usually not directly identifiable but some research in de-anonymizing bitcoin users have shown that users can be identified successfully. As a good practice it is suggested that users generate a new address for each transaction in order to avoid linking transactions to the common owner, thus avoiding 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 some other elements such as the previous block hash (hash pointer), timestamp, and nonce.

Peer-to-peer network

As the name implies, this is a network topology whereby all peers can communicate with each other and send and receive messages.

Scripting or programming language

This element performs various operations on a transaction. Transaction scripts are predefined sets of commands for nodes to transfer tokens from one address to another and perform various other functions. Turing complete programming language is a desirable feature of blockchains; however, the security of such languages is a key question and an area of important and ongoing research.

Virtual machine

This is an extension of a transaction script. A virtual machine allows Turing complete code to be run on a blockchain (as smart contracts) whereas a transaction script can be limited in its operation. Virtual machines are not available on all blockchains; however, various blockchains use virtual machines to run programs, for example **Ethereum Virtual Machine** (**EVM**) and **Chain Virtual Machine** (**CVM**).

State machine

A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next and eventually to a final form as a result of a transaction execution and validation process by nodes.

Nodes

A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This is done by following a consensus protocol. (Most commonly this is PoW.) Nodes can also perform other functions such as simple payment verification (lightweight nodes), validators, and many others functions depending on the type of the blockchain used and the role assigned to the node.

Smart contracts

These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met. The smart contract feature is not available in all blockchains but is now becoming a very desirable feature due to the flexibility and power it provides to the blockchain applications.

Features of a blockchain

A blockchain performs various functions. These are described below in detail.

Distributed consensus

Distributed consensus is the major underpinning of a blockchain. This enables a blockchain to present a single version of truth that is agreed upon by all parties without the requirement of a central authority.

Transaction verification

Any transactions posted from nodes on the blockchain are verified based on a predetermined set of rules and only valid transactions are selected for inclusion in a block.

Platforms for smart contracts

A blockchain is a platform where programs can run that execute business logic on behalf of the users. As explained earlier, not all blockchains have a mechanism to execute smart contracts; however, this is now a very desirable feature.

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.

Generating cryptocurrency

This is an optional feature depending on the type of blockchain used. A blockchain can generate cryptocurrency as an incentive to its miners who validate the transactions and spend resources in order to secure the blockchain.

Smart property

For the first time it is possible to link a digital or physical asset to the blockchain in an irrevocable manner, such 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 it with a digital music file, for example, which can be copied many times without any control; on a blockchain, however, if you own it no one else can claim it unless you decide to transfer it to someone. This feature has far-reaching implications especially in **Digital Rights Management** (**DRM**) and electronic cash systems where double spend detection is a key requirement. The double spend problem was first solved in bitcoin.

Provider of security

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 has become a main barrier for its adaptability by financial institutions and other industries that need privacy and confidentiality of transactions. As such it is being researched very actively and there is already some good progress made. It could be argued that in many situations confidentiality is not really needed and transparency is preferred instead. For example, in bitcoin confidentiality is not really required; however, it is desirable in some scenarios. Research in this area is very ripe and already major progress has been made towards providing confidentiality and privacy on blockchain. A more recent example is Zcash, which will be discussed in more detail in later chapters. Other security services such as nonrepudiation and authentication are also provided by blockchain as all actions are secured by using private keys and digital signatures.

Immutability

This is another key feature of blockchain: records once added onto the blockchain are immutable. There is the possibility of rolling back the changes but this is considered almost impossible to do as it will require an unaffordable amount of computing resources. For example, in much desirable case of bitcoin if a malicious user wants to alter the previous blocks then it would require computing the PoW again for all those blocks that have already been added to the blockchain. This difficulty makes the records on a blockchain practically immutable.

Uniqueness

This feature of blockchain ensures that every transaction is unique and has not been spent already. This is especially relevant in cryptocurrencies where much desirable detection and avoidance of double spending are a key requirement.

Smart contracts

Blockchain provides a platform to run smart contracts. These are automated autonomous programs that reside on the blockchain and encapsulate business logic and code in order to execute a required function when certain conditions are met. This is indeed a revolutionary feature of blockchain as it allows flexibility, programmability, and much desirable control of actions that users of blockchain need to perform according to their specific business requirements.

Applications of blockchain technology

Blockchain technology has a multitude of applications in various sectors including but not limited to finance, government, media, law, and arts. More light will be shed on these aspects in `Chapter 9`, *Hyperledger* where practical use cases will be discussed in detail for various industries. It is sufficient to say for now that almost all industries have already realized the potential and promise of blockchain and have already embarked, or soon will embark, on the journey to benefit from the blockchain technology.

In the following section, a general scheme of creating blocks is discussed. This is presented here to give readers a general idea of how blocks are generated and what the relationship is between transactions and blocks.

How blockchains accumulate blocks

1. A node starts a transaction by signing it with its private key.
2. The transaction is propagated (flooded) by using much desirable Gossip protocol to peers, which validates the transaction based on pre-set criteria. Usually, more than one node is required to validate the transactions.
3. Once the transaction is validated, it is included in a block, which is then propagated on to 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.
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.

Steps 4 and 5 can be 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 steps 4 and 5.

Tiers of blockchain technology

In this section, various tiers of blockchain technology are discussed. It is envisaged that, due to the rapid development and progress made in blockchain technology, many applications will evolve over time. Some have already been realized while some can be envisioned for the future based on the current rate of advancement in the blockchain technology.

First, the three levels discussed below were originally described by *Melanie Swan* in her book *Blockchain, Blueprint for a New Economy* as tiers of blockchain categorized on the basis of applications in each category. In addition to this, Tier X or Generation X is discussed later. This is what the author thinks will become a reality when the blockchain technology becomes advanced enough.

Blockchain 1.0

This was introduced with the invention of bitcoin and is basically used for cryptocurrencies. Also, as bitcoin was the first implementation of cryptocurrencies it makes sense to categorize Generation 1 of blockchain technology to only include cryptographic currencies. All alternative coins and bitcoin fall into this category. This includes core applications such as payments and applications.

Blockchain 2.0

Generation 2.0 blockchains are used by financial services and contracts are introduced in this generation. This includes various financial assets, for example derivatives, options, swaps, and bonds. Applications that are beyond currency, finance, and markets are included at this tier.

Blockchain 3.0

Generation 3 blockchains are used to implement applications beyond the financial services industry and are used in more general-purpose industries such as government, health, media, the arts, and justice.

Generation X (Blockchain X)

This is a vision of blockchain singularity where one day we will have a public blockchain service available that anyone can use just like the Google search engine. It will provide services in all realms of society. This is a public open distributed ledger with general-purpose rational agents (Machina Economicus) running on blockchain, making decisions and interacting with other intelligent autonomous agents on behalf of humans and regulated by code instead of law or paper contracts. This will be elaborated in detail in `Chapter 13`, *Current Landscape and What's Next*.

Types of blockchain

Based on the way blockchain has evolved over the last few years, it can be divided into multiple types with distinct but sometimes partly overlapping attributes.

Public blockchains

As the name suggests, these blockchains 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. These ledgers are not owned by anyone and are publicly open for anyone to participate in. All users of the permission-less ledger maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism in order to reach a decision about the eventual state of the ledger. These blockchains are also known as permission-less ledgers.

Private blockchains

Private blockchains as the name implies are private and are open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves.

Semi-private blockchains

Here part of the blockchain is private and part of it is public. The private part is controlled by a group of individuals whereas the public part is open for participation by anyone.

Sidechains

More precisely known as pegged sidechains, this is a concept whereby coins can be moved from one blockchain to another and moved back. Common uses include the creation of new altcoins (alternative cryptocurrencies) whereby coins are *burnt* as a proof of adequate stake. There are two types of sidechain. The example provided above for *burning* coins is applicable 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.

Permissioned ledger

A permissioned ledger is a blockchain whereby the participants of the network are known and already trusted. Permissioned ledgers do not need to use a distributed consensus mechanism, instead an *agreement protocol* can be used to maintain a shared version of truth about the state of the records on the blockchain. There is also no requirement for a permissioned blockchain to be private as it can be a public blockchain but with regulated access control.

Distributed ledger

As the name suggests, this ledger is distributed among its participants and spread across multiple sites or organizations. This type can either be private or public. The key idea is that, unlike many other blockchains, the records are stored contiguously instead of sorted into blocks. This concept is used in Ripple.

Shared ledger

This is generic term that is used to describe any application or database that is shared by the public or a consortium.

Fully private and proprietary blockchains

These blockchains perhaps have no mainstream application as they deviate from the core idea of decentralization in blockchain technology. Nonetheless in specific private settings within an organization there might be a need to share data and provide some level of guarantee of the authenticity of the data. These blockchains could be useful in that scenario. For example, for collaboration and sharing data between various government departments.

Tokenized blockchains

These blockchains are standard blockchains that generate cryptocurrency as a result of a consensus process via mining or via initial distribution.

Tokenless blockchains

These are probably not real blockchains because they lack the basic unit of transfer of value but are still valuable in situations where there is no need to transfer value between nodes and only sharing some data among various already trusted parties is required.

In the next section, the idea of consensus from a blockchain perspective will be discussed. Consensus is the backbone of a blockchain and provides decentralization of control as a result through an optional process known as mining. The choice of consensus algorithm is also governed by the type of blockchain in use. Not all consensus mechanisms are suitable for all types of blockchains. For example, in public permission-less blockchains it would make sense to use PoW instead of some basic agreement mechanism that perhaps is based on proof of authority. Therefore it is essential to choose a consensus algorithm appropriately for a blockchain project.

Consensus in blockchain

Consensus is basically a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of truth by all peers on the blockchain network. This concept was discussed in the distributed systems section earlier in this chapter.

Roughly, the following two categories of consensus mechanism exist:

1. Proof-based, leader-based, or the *Nakamoto consensus* whereby a leader is elected and proposes a final value
2. Byzantine fault tolerance-based, which is a more traditional approach based on rounds of votes

Consensus algorithms that are available today or are being researched in the context of blockchain are presented later. This is not an exhaustive list but an attempt has been made to present all important algorithms.

Proof of Work

This type of consensus mechanism relies on proof that enough computational resources have been spent before proposing a value for acceptance by the network. This is used in bitcoin and other cryptocurrencies. Currently, this is the only algorithm that has proven astonishingly successful against Sybil attacks.

Proof of Stake

This algorithm works on the idea that a node or user has enough stake in the system; for example the user has invested enough in the system so that any malicious attempt would outweigh the benefits of performing an attack on the system. This idea was first introduced by Peercoin and is going to be used in the Ethereum blockchain. Another important concept in **Proof of Stake** (**PoS**) is coin age, which is a derived from the amount of time and the 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

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

Proof of Elapsed Time

Introduced by Intel, it uses **Trusted Execution Environment** (**TEE**) to provide randomness and safety in the leader election process via a guaranteed wait time. It requires the Intel **SGX** (**Software Guard Extensions**) processor in order to provide the security guarantee and for it to be secure. This concept is discussed in more detail in `Chapter 9`, *Hyperledger* in the context of the Intel Sawtooth Lake blockchain project.

Deposit-based consensus

Nodes that wish to participate on the network have to put in a security deposit before they can propose a block.

Proof of importance

This idea is important and different from Proof of Stake. Proof of importance not only relies on how much stake a user has in the system but it also monitors the usage and movement of tokens by the user to establish a level of trust and importance. This is used in Nemcoin.

Federated consensus or federated Byzantine consensus

Used in the stellar consensus protocol, nodes in this protocol keep a group of publicly trusted peers and propagates only those transactions that have been validated by the majority of trusted nodes.

Reputation-based mechanisms

As the name suggests, a leader is elected on the basis of the reputation it has built over time on the network. This can be based on the voting from other members.

Practical Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance (**PBFT**) achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols, including but are not limited to 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.

CAP theorem and blockchain

Strangely, it seems that the CAP theorem is violated in blockchain, and especially in the most successful implementation: bitcoin, but 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. For this purpose, the concept of mining was introduced in bitcoin; this is a process that facilitates the achievement of consensus by using a consensus algorithm called PoW. At a higher level, mining can be defined as a process that is used to add more blocks to the blockchain.

Benefits and limitations of blockchain

Numerous benefits of blockchain technology are being discussed in the industry and proposed by thought leaders around the world in blockchain space. The top 10 benefits are listed and discussed as follows.

Decentralization

This is a core concept and benefit of 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

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

Immutability

Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not truly immutable but, due to the fact that changing data is extremely difficult and almost 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 each and every node, the system becomes highly available. Even if nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available.

Highly secure

All transactions on a blockchain are cryptographically secured and provide integrity.

Simplification of current paradigms

The current model in many industries such as finance or health is rather disorganized, wherein multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. But as a blockchain can serve as a single shared ledger among interested parties, this can result in simplifying this 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 allowing the quicker settlement of trades as it 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 third party or clearing houses are required in the blockchain model, this can massively eliminate overhead costs in the form of fees that are paid to clearing houses or trusted third parties.

Challenges and limitations of blockchain technology

As with any technology there are challenges that need to be addressed in order to make a system more robust, useful, and accessible. Blockchain technology is no exception; in fact a lot of effort is being made in Academia and Industry to overcome the challenges posed by blockchain technology. A selection of the most sensitive challenges are presented as follows:

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

All these and more will be discussed in detail with possible solutions in `Chapter 13`, *Current Landscape and What's Next*.

This chapter has been kept generic and less technical on purpose. Once cryptography has been explained in detail in `Chapter 3`, *Cryptography and Technical Foundations*, specific blockchain solutions will be discussed in appropriate technical depth and detail.

Summary

This chapter introduced blockchain technology at a high level to the readers. First some basic ideas regarding distributed systems were discussed then the history of blockchain was introduced. Concepts such as electronic cash and hashcash were discussed. Furthermore, various definitions of blockchain from different points of views were presented. Some applications of blockchain technology were also discussed briefly. Next in the chapter, different types of blockchain were introduced. Finally, the benefits and limitations of this new technology were also introduced. Some topics were introduced only lightly on purpose as they will be discussed in depth in later chapters. For example, challenges and limitations were only mentioned in the chapter but no details were provided as there is a full chapter dedicated to this later in the book. In the next chapter, readers will be introduced to the concept of decentralization, which is central to the concept of blockchains and their vast number of applications.