LearnThings.Online

Subscribe

Blockchain: Understanding Its Uses and Implications – Chapter 1. Introduction to Blockchain

This course is from edX

Scroll down click “Read More” to check original post on edX.

Blockchain technology is changing how business is executed. It’s important to understand why blockchain is different and how it works in comparison with technologies of the past.

The first segment of this course covers all the main concepts of what Blockchain is. It discusses how it began as a triple ledger system first introduced for the administration of the cryptocurrency Bitcoin, and how it is now applied to all aspects of business including government, banking, supply chains, and a host of other industries.

It also analyzes the concept of transparent ledgers, both public and permissioned, and focuses on using cryptography to achieve consensus, immutability, and governance of transactions. This is all part of Blockchain’s ability to provide “trusted data from untrusted sources,” disrupting traditional accounting methodologies and international trade.

The course then dives into the various methods of blockchain governance that currently exist in the marketplace as well as how consensus fits into governance. It explores how to reach consensus through proof-of-work or

proof-of-stake.

Other aspects of the course include examining the very specific features of blockchain that solve problems that have been difficult to overcome in the past with more centralized architectures.

The final part of the course takes a deep dive into the various use cases of blockchain, complete with analyzing real examples of how different industries are executing the technology and improving their business. Examining a problem, and then depicting a blockchain use case that solves the problem, will help gain an understanding of how blockchain is applied to real-world situations.

Syllabus

Welcome & Introduction

Chapter 1. Introduction to Blockchain

This section covers some of the technical aspects that comprise a blockchain and explain why blockchain is different and “works” in comparison with technologies of the past.

Chapter 2. Governance and Consensus

This section covers the various methods of blockchain governance that currently exist in the marketplace as well as how consensus fits into governance. It also covers various levels of governance and how it works with both public and permissioned blockchains.

Chapter 3. Blockchain Problem Solving

This section takes a look at the very specific features of blockchain that solve problems that have been difficult to solve in the past with more centralized architectures.

Chapter 4. Blockchain Use Cases

This section covers various use cases of blockchain. It examines the problem, and then depicts a blockchain use case that solves the problem.

Final Exam

Check other chapters if you finish this chapter.

- Chapter 1
- Chapter 2
- Chapter 3
- Chapter 4

Chapter 1. Introduction to Blockchain

Chapter 1: Learning Objectives

By the end of this chapter, you should be able to:

- Discuss about blockchain, its characteristics and uses.
- Discuss about the roles and users in a blockchain.
- Explain how blockchain is using cryptography.

- Discuss about public and private blockchains: differences, advantages/disadvantages, use cases.
- Explain what consensus is and why consensus is needed in a blockchain.

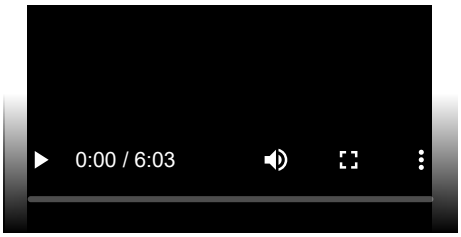
Introduction to Blockchain

Learning Outcomes

By the end of this section, you should be able to:

- Explain the concept of blockchain.
- Discuss the importance of blockchain.
- Understand how bitcoin is using blockchain.

Video: What Is Blockchain?



Blockchain and the Early Internet

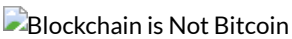
Let's Recap: Comparing Blockchain to the Early Internet

- When the internet first came along, we had no idea how it would forever change our lives.
- From smart phones and text messages to streaming movies and FaceTime visits with loved ones, no one knew the ways the world would change with the invention of the Internet.
- We are currently in the early phases of blockchain and there is much potential yet to be unlocked.

Blockchain Is NOT Bitcoin

Blockchain is NOT Bitcoin:

- Bitcoin is transacted over an open, public, anonymous blockchain network.
- Bitcoin and cryptocurrencies are great use cases for blockchain, but there are many more use cases that utilize blockchain technology.



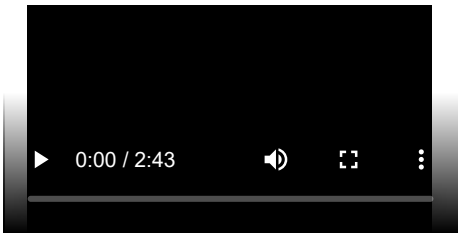
Basics of Blockchain

Learning Outcomes

By the end of this section, you should be able to:

- Explain what the block in blockchain is.
- Explain how blocks are chained together.
- Discuss the concept of immutability in a blockchain.
- Describe users and their roles in a blockchain.

Video: Blockchain – Let's Cover the Basics



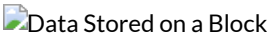
Let's Review an Analogy

A block on a blockchain can be thought of much like a page in a notebook. Data is stored on a block, just like data is written on a page of a notebook.

Data Stored

Any data can be stored on the same block. Examples of stored data include:

- Medical Records
- Property Agreements
- Voting.



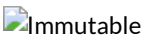
Subscribe

Blocks Are Chained Together

Each block is chained or tied to the previous block by embedding the block with information from the previous block (we will go through this in depth later in the course).

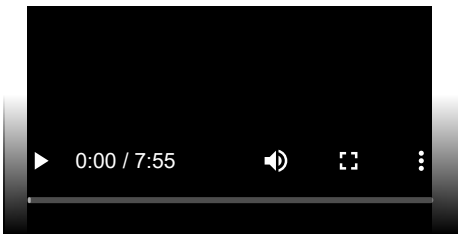
Blockchain Is Immutable

If the data is tampered with anywhere in the chain, the links will break in a very obvious way:



This provides immutability and security.

Video: Introduction to Blockchain Components



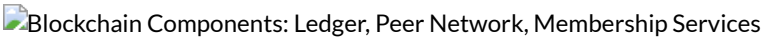
Blockchain Components

The blockchain is built of several different types of components, each with a specific role to play within the blockchain's operation:

- **Ledger**: A distributed, immutable historical record
- **Peer Network**: Stores, updates, and maintains the ledger
- **Membership Services**: User authentication, authorization, and identity management
- **Smart Contract**: Program that runs on the blockchain
- **Wallet**: Stores users' credentials
- **Events**: Notifications of updates and actions on the blockchain
- **Systems Management**: Component creation, modification, and monitoring
- **Systems Integration**: Integration of blockchain with external systems.

Next, let's discuss each of these components in more detail.

Blockchain Components: Ledger, Peer Network, and Membership Services



Ledger: A distributed, immutable historical record
The goal of the blockchain is to create a distributed, immutable record of the history of the blockchain called the ledger.

Peer Network: Stores, updates, and maintains the ledger
The ledger is stored, updated, and maintained by a peer network. Each node in this network maintains its own copy of the ledger. It is the job of the network as a whole to come to a consensus on the contents of each update to the ledger. This ensures that each individual copy of the ledger is identical without requiring a centralized "official" copy of the ledger.

Membership Services: User authentication, authorization, and identity management
On some blockchains, anyone can join the peer network and all network members have equal powers and authority. Permissioned blockchains require authorization to join and Membership Services authenticates, authorizes, and manages the identity of users on the blockchain.

Blockchain Components: Smart Contracts, Wallet and Events

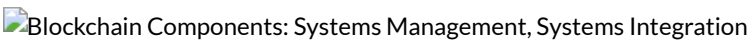
Smart Contract: Program that runs on the blockchain
The original blockchains were designed to simply allow financial transactions to be performed and stored in the historical ledger, and had limited configurability. Since then, blockchains have evolved so that some have become

fully functional distributed computers. Smart contracts are programs that run on the blockchain. Users can interact with smart contracts in a similar way that they interact with programs on a standard computer.

Wallet: Stores users credentials
In blockchain, the user's wallet stores their credentials and tracks digital assets associated with the user's address. The wallet tracks user credentials and any other information that may be associated with their account.

Events: Notifications of updates and actions on the blockchain
The blockchain's ledger and the state of the peer network are updated by events. Examples of events include the creation and dispersion of a new transaction across the peer network and the addition of a new block to the blockchain. Events may also include notifications from smart contracts on blockchains that support such contracts.

Blockchain Components: Systems Management and Integration

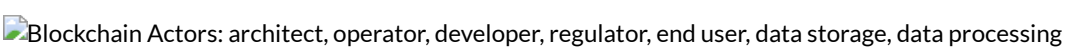


Systems Management: Component creation, modification, and monitoring
The blockchain is designed to be a long-lived system in a field that is constantly evolving. Systems management provides the capability of creating, modifying, and monitoring blockchain components to meet the needs of its users.

Systems Integration: Integration of blockchain with external systems
As blockchain has evolved and increased in functionality, it has become more common to integrate blockchains with other external systems, commonly through the use of smart contracts. While this is not a specific component of the blockchain, systems integration is included to acknowledge this capability.

Blockchain Actors

A business blockchain solution requires many actors playing a variety of roles to be fully functional:



Blockchain Actors: Architect

A **Blockchain Architect** is the designer of the blockchain solution. For a blockchain solution to be functional, it first needs to exist. The blockchain architect is the person or group who design the blockchain.

Blockchain Actors: Operator

The **Blockchain Operator** stores, maintains, and updates the blockchain ledger. Once the blockchain solution is designed and built, an operator can join to create the peer network mentioned previously. The role of the operator is to set up and maintain peers within the network.

Blockchain Actors: Developer

The **Blockchain Developer** creates smart contracts to run on the blockchain. The functionality of the blockchain has been greatly expanded by the introduction of blockchains that support smart contracts. Developers design and upload smart contracts to the blockchain to expand its capabilities. In addition to implementing the

smart contracts, you may also have front-end developers who implement applications that access the blockchain (i.e., the applications initiate the transactions on the blockchain).

Blockchain Actors: Regulator

The **Blockchain Regulator**: Many businesses operate under regulations regarding how their data should be stored and processed. For blockchain solutions, a regulator may have greater visibility into the historical ledger due to their role within the organization.

Blockchain Actors: End User

The **End User** is the consumer of services built around the blockchain. Typically, this involves using software that uses the blockchain as a backend storage solution. Users rarely interact directly with the blockchain.

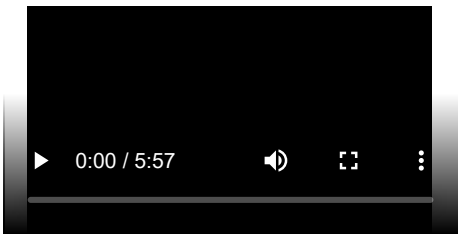
Blockchain Actors: Data Storage

Data Storage is represented by traditional databases for storing data off-chain. The blockchain provides distributed, immutable storage with built-in integrity checking; however, it has a maximum capacity based on the standard block size and block rate. To provide integrity verification for large amounts of data, it is common to store the data off-chain and store a hash of the data on-chain. This guarantees that the data is not being modified while protecting the blockchain from becoming bloated.

Blockchain Actors: Data Processing

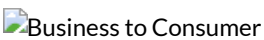
Data Processing is represented by an external system used for additional processing. Smart contracts execute on the blockchain, meaning that each member of the peer network must execute the code to remain in sync with the current state of the network. If smart contracts commonly require large amounts of processing power to complete, devices external to the peer network may be used to augment the processing power of the network.

Video: Who Is Using Blockchain?



Blockchain Users: Business to Consumer (B2C)

A **Business to Consumer** (B2C) is business or transactions conducted directly between a company and consumers who are the end-users of its products or services. Services being provided to the consumer is an area of interest to companies.



B2C Transparency

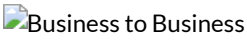
Who is Using Blockchain: The Jewelry Industry

The jewelry industry has been known for fraud, child labor issues, false metal mining, and a clear lack of transparency. A precious metals consortium with IBM has established a blockchain initiative around how transparency can be brought to the consumer:

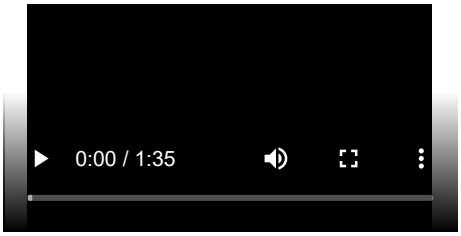
- The TrustChain™ Initiative tracks and authenticates diamonds and precious metals through every stage of the supply chain as it becomes a piece of finished jewelry.
- It provides digital verification, physical product and process verification, as well as third-party oversight. The collaboration's goal is to instill trust in the origin and ethical sourcing of jewelry by bringing together a community of responsible and ethical organizations across the complex and multi-tiered jewelry supply chain.
- Consumers will see that TrustChain™ establishes a trusted product with documented provenance and brings together quality assurance, social and environmental responsibility, and authenticity spanning the entire jewelry ecosystem – from miners, manufacturers, wholesale suppliers, and retailers – on a single digital platform.

Blockchain Users: Business to Business (B2B)

Business to Business (B2B) in blockchain:



Video: Dubai Use Case



Subscribe

Dubai and Blockchain

Dubai aims to be a pioneer in the adoption of emerging technologies such as blockchain, which it recognizes has a major potential to transform city services. Dubai is one of the first to fully implement blockchain on a city-wide basis.

Smart Dubai 2021

Dubai is investing in the Smart Dubai Office (SDO) and 1776 Launch Blockchain Challenge. Sponsored by His Highness Sheikh Hamdan, Dubai is funding blockchain implementation at many levels.

The Dubai Blockchain Strategy is based on three pillars:

- **Government Efficiency**: Implementing blockchain technology in government services
- **Industry Creation**: Supporting the creation of a blockchain industry through empowering startups and businesses
- **Thought Leadership**: Leading the global thinking on blockchain technology.

Dubai's adoption of blockchain technology at a city-wide scale is a testament to its commitment to positively transform government from service provider to service enabler.

Distributed Ledgers

Learning Outcomes

By the end of this section, you should be able to:

- Describe single, double, and triple entry accounting models.

- Recall the short history of a distributed ledger.
- Summarize why blockchain is the modern day distributed ledger.

Understanding Ledgers

History of Ledgers: Single-Entry Ledgers

- Ledgers first appeared around 3,000 B.C.
- Single-entry only.
- Chanakya, an Indian leader, creates the first documented accounting standards.

History of Ledgers: Double-Entry Accounting

- Double-entry ledger appears in 1340 A.D.
- Tracks debits and credits.
- Tells the story of a transaction from both/all sides.
- The Italian Luca Pacioli, recognized as the father of accounting and bookkeeping, was the first person to publish a work on double-entry bookkeeping and introduced the field in Italy.

History of Ledgers: Triple-Entry Accounting

- Triple-entry accounting is an enhancement to the traditional double-entry system, in which all accounting entries involving outside parties are cryptographically sealed by a third entry.
- Debits, credits, and an immutable link to all past debits and credits.
- Triple-entry ledger appears in 2008 in a white paper by Satoshi Nakamoto (a.k.a., Blockchain).

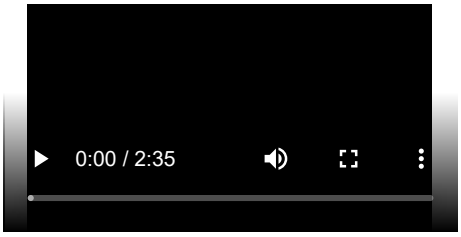
Let's review an example: A seller books a debit to account for cash received, while a buyer books a credit for cash spent in the same transaction, but in separate sets of accounting records. This is where the blockchain comes in: rather than these entries occurring separately in independent sets of books, they occur in the form of a transfer between wallet addresses in the same distributed, public ledger, which creates an interlocking system of enduring accounting records. Since the entries are distributed and cryptographically sealed, falsifying them in a credible way or destroying them to conceal activity is practically impossible.

Triple-Entry Accounting Features:

> - Tamper-Proof Records
> - Distributed Ledgers
> - Double-Entry+Cryptography
> - Validated, Secure, and Private
> - Digitally Signed Receipts.

Double vs. Triple-Entry Accounting Example

Video: Island of Yap



Subscribe

Island of Yap: Recap

Island of Yap: An Example

Let's review an example on Yap Island:

- Alice agrees to trade Bob her stone by the pond in exchange for all of his cattle.
- Alice and Bob announce their transaction to the tribe.
- Everyone updates their mental ledger.
- From this point on, they agree that the stone by the pond is owned by Bob.

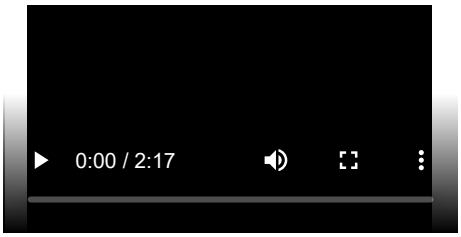
Island of Yap: Corruption Is Near Impossible

Alice tries to corrupt Carol, so that Carol's ledger shows that Alice never gave up ownership of the stone.

- **Centralized ledger**:
  Only one place to cheat.
- **Decentralized ledger**:
  Carol will be outvoted by the rest of the tribe, and her version of the ledger will not be accepted.

If Alice wants to cheat, she will need a way to convince 51% or more of the tribe to accept an alternative ledger.

Video: The Evolution of Distributed Ledgers



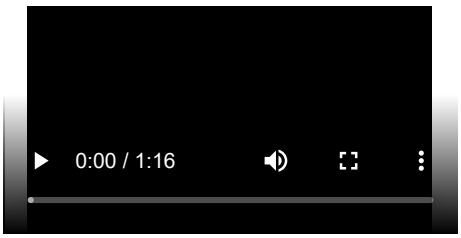
Cryptography

Learning Outcomes

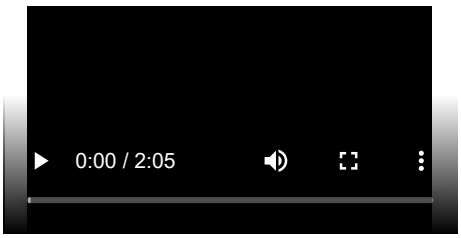
By the end of this section, you should be able to:

- Discuss how cryptography allows for distributed ledgers to work on a global scale.
- Explain how blockchain is using cryptography.
- Define key terms of cryptography and understand its basics.
- Discuss cryptography types: public/private key, hash functions.
- Explain the concept of Merkle trees and how they are used in blockchain technologies.
- Discuss how different blockchains use cryptography.

Video: From Distributed Ledger to Cryptography



Video: Rose Greenhow



Cryptography and Hashing in Blockchain

- Blockchain provides users with data integrity in a trustless environment.
- This is accomplished using cryptography in a way that moves the burden of trust from data processors to cryptographic algorithms.

In this section, we will discuss some of the ways cryptography is used in the blockchain.

Annotate

Cryptography Key Terms

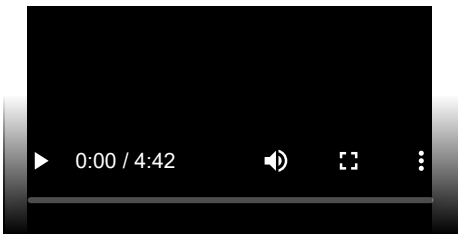
Let's explain the key terms used in cryptography:

- **Secret**: The data which we are trying to protect
- **Key**: A piece of data used for encrypting and decrypting the secret
- **Function**: The process or function used to encrypt the secret
- **Cipher**: The encrypted secret data, the output of the function.

The Secret and the Key are passed into the Function to create the Cipher.

Subscribe

Video: Cryptography Basics



Cryptographic Function

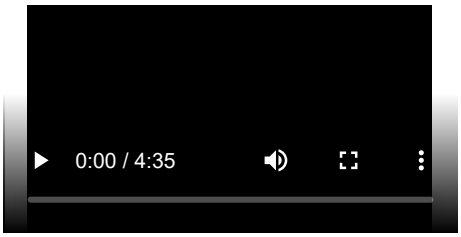
What is a cryptographic function?

- A function for encoding or encrypting data to protect the contents from adversaries.

Simple example function:

- Secret = “Blockchain Training Alliance”
- Function = Swap each letter in the secret with a new letter according to the Key
- Key = “+2”
- Cipher = “Dnqemejckp Vtckpcpi Cnnkcpeg”.

Video: Byzantine Fault Tolerance



Subscribe

Types of Cryptography

Blockchain makes use of several different types of cryptography. Among these are:

- **Public Key Cryptography**
  Pair of public and private keys used for encryption and digital signatures.
- **Zero-Knowledge Proof**
  Prove knowledge of a secret without revealing it.
- **Hash Functions**
  One-way pseudo-random mathematical functions and Merkle trees.

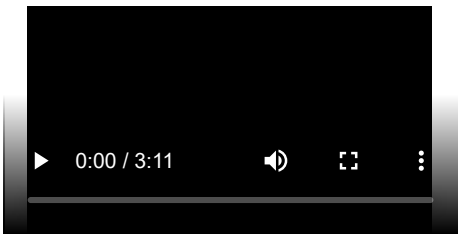
Public Key Cryptography

Public key cryptography uses a pair of a public key and a private key to perform different tasks. Public keys are widely distributed, while private keys are kept secret.

Using a person’s public key, it is possible to encrypt a message so that only the person with the private key can decrypt and read it. Using a private key, a digital signature can be created so that anyone with the corresponding public key can verify that the message was created by the owner of the private key and was not modified since.

Blockchain makes extensive use of public key cryptography.

Video: Private/Public Key Cryptography

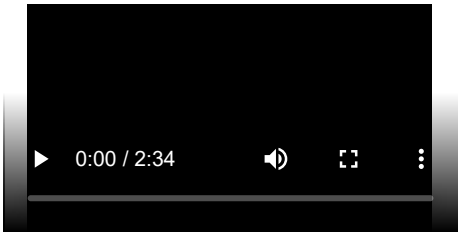


Zero-Knowledge Proof

Zero-knowledge proof is the ability to prove a secret without revealing what the secret is.

Let's review an example: Let's say there are two toy cars, identical in shape and size, except, one is red and one is blue. Jerry, who is color-blind, holds the toy cars behind his back. Jerry then shows one of the cars to Sam. Jerry then hides that car behind his back and shows Sam the other car. Sam can consistently detect the switch because the cars are different colors, but he never has to reveal the color of the cars to Jerry in order to prove the secret.

Video: Zero-Knowledge Proof – Cave Example

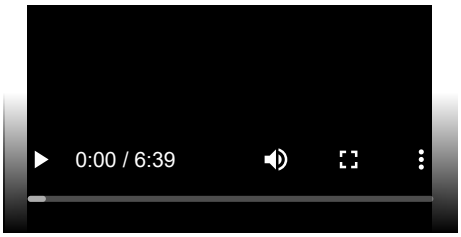


Hash Functions

Finally, hash functions feature heavily in blockchain. A hash function is a mathematical equation with four important properties:

- Hash functions can take anything as input and create an output with a fixed size. This makes it possible to condense anything into a piece of data of a fixed size and is how messages are condensed for digital signatures.
- It's easy to calculate a hash, but hard to determine a hash input from the output. The best option is to keep trying inputs until one produces the desired output.
- Inputs that differ by a single bit produce hashes that differ by half of their bits on average. This prevents someone from finding a desired hash input using a "hill climbing".
- It is infeasible to find two inputs that produce the same output when hashed. Since a hash can take any input and produce a fixed output, it makes sense that multiple different inputs will create the same output. A good hash function will make it so that you have to try a large number of inputs before finding two that produce the same output.

Video: Cryptographic Hashing Demo



Lab 1: Hashing

Next, let's engage with an interactive lab. This lab is an actual hands-on demonstration of taking data and creating a hash output. Enjoy!

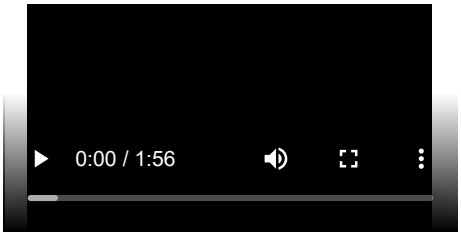
Start Lab

Merkle Tree

A special type of data storage structure based on hash functions is called a **Merkle tree**:

- It is structured as a binary tree; the leaves contain the values to be stored and each internal node is the hash of its two children.
- It provides efficient lookups and protection against forgery since verifying a transaction is included in the tree. Can be accomplished by sending only the transaction, the hash contained in each node between the transaction leaf node and the root, and the hash values used to create each hash sent.
- Looking up a transaction in a Merkle tree with three levels includes sending two transactions (the desired one and the other child of its parent) and three hashes (the transaction's parent, the root, and the root's other child).

Video: Merkle Tree, Validation of Data



Comparative Cryptography Usage

Ethereum and Hyperledger Fabric are two blockchain systems that are taking different approaches to moving business to the blockchain. On the next few pages, we will compare how Ethereum and Hyperledger Fabric make use of:

- Public key cryptography
- Zero-knowledge proofs
- Hash functions.

Ethereum vs. Hyperledger Fabric: Cryptography Usage

The use of public keys for identity management is a logical choice since knowledge of a public key is necessary for verification of digital signatures. Both Ethereum and Hyperledger Fabric use digital signatures on transactions and blocks to verify the identity of the creator and that the signed data has not been modified since signing. Public key cryptography is used in the blockchain as a method for managing users' identities without revealing real world identities.

In Ethereum, users are identified by an address that is directly related to the user's public key. This provides identity verification while preserving anonymity.

In Hyperledger Fabric, users are identified via X.509 certificates. These certificates provide several pieces of information about the user, but one of these is also the user's public key.

Zero-knowledge proofs are a cryptographic principle used in some blockchains to increase the privacy of users. Currently, Ethereum does not have support for zero-knowledge proofs, but adding the necessary functionality for zkSNARKS, a type of zero-knowledge proof, is currently included in the Ethereum development roadmap. Hyperledger Fabric does not currently support zero-knowledge proofs as a privacy feature.

Ethereum vs. Hyperledger Fabric: Hashing Usage

Hash functions are at the core of all blockchain technology. One of the primary uses for hash functions is chaining blocks together. In both Ethereum and Hyperledger Fabric, blocks include the hash of the previous block to tie the blockchain into a cohesive whole.

Merkle trees are a data structure that allows authenticated storage with efficient data retrieval. Both Ethereum and Hyperledger Fabric are smart contract platforms that use a particular type of Merkle tree called the Patricia tree to store the current state of their virtual machine.

Hash functions are used as the cryptographic puzzle at the center of the Proof of Work consensus algorithm. Ethereum currently uses Proof of Work for consensus, though a switch to Proof of Stake has been built into the road map from the beginning. There are only two consensus algorithms implemented in Hyperledger Fabric – Solo and Kafka. SOLO is for development and Kafka is for production.

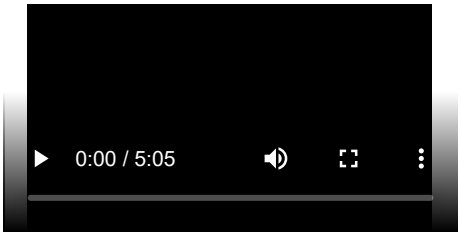
Transparency

Learning Outcomes

By the end of this section, you should be able to:

- Outline the advantages of transparency.
- Discuss how a blockchain utilizes transparency.

Video: Disintermediation – Trust Through Transparency



Traditional Database Updates

Traditional databases using the CRUD update model have four main operations:

Transparency of Traditional Databases

Traditional databases do not retain historical information:

- Only the most recent versions of each value is visible.
- Deleted values are not visible in the database.

This limits the transparency of data contained in the database:

- Values can be modified or deleted after creation.

The CRUD update model of databases allows data to be changed or removed from the database. This means that the visible data in a database is not an accurate historical record of the database. The existence of an Update operation means that each value in the database is only the most recent version of that value and could have had different values in the past. The Delete operation means that values can be removed from the database.

This limits the transparency of data in the database since values can be modified or deleted after creation.

Blockchain Updates

The blockchain is designed to be a data structure that only allows appending:

- The past history of the blockchain is visible and immutable.
- Updates to the blockchain can be performed by including them in new blocks added to the blockchain.

The blockchain is designed as a data structure where each block in the chain locks in the value of the previous block and so on, back to the first or *genesis* block. This means that the blockchain is an append-only data structure without support for modification or deletion.

The entire history of the blockchain is publicly visible and stored in a distributed and decentralized fashion. Values in the blockchain can be "updated" by appending a new version of that value in a later block, but the complete history of the value is preserved.

Transparency of a Blockchain

The blockchain is designed so that its entire history is visible and unchangeable. Transactions in the blockchain cannot be modified after creation, and their complete history is publicly visible. This means that the blockchain is a completely transparent data structure with the useful property that the integrity of the blockchain is easily verifiable by any user.

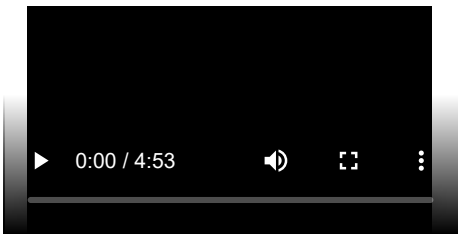
Transparency: Databases vs. Blockchain

Traditional databases and the blockchain were created for different purposes and have different levels of transparency. Traditional databases have low transparency since values can be modified or deleted; however, this changeability allows them to store data in an efficient manner, with only the most relevant versions of each value retained in storage.

The blockchain is publicly visible and immutable, meaning that it has very high transparency. Its append-only structure and decentralized storage sacrifice storage efficiency for trustworthiness of the stored data.

Annotate

Video: Transparency



Subscribe

Immutability

Learning Outcomes

By the end of this section, you should be able to:

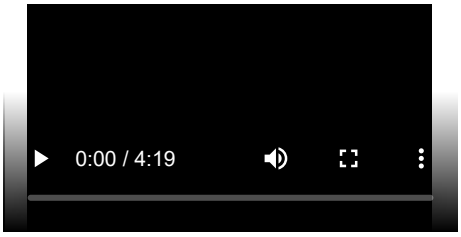
- Explain what it means to be immutable and how blockchain is immutable.
- Outline the advantages of immutability.
- Discuss how chaining provides immutability.

Immutability in the Blockchain

Blockchain is designed to be an authoritative ledger of the history of the network.

This history may include financial transactions and business agreements where modifications to the ledger may have wide-reaching business impacts. Blockchain is based on an untrusted network, so trust that the blockchain has not been modified needs to arise from the structure of the blockchain itself, rather than from trust in the organization storing a certain copy.

Video: Immutability



Immutability in the Blockchain Is Essential

The blockchain needs to be immutable. If someone can change the blockchain after the fact, then it is no longer a trusted historical ledger. The blockchain is designed so that immutability is cumulative; each piece is linked to every other piece, creating a cohesive whole that is more difficult for an attacker to modify.

- At the bottom level, transactions are digitally signed by their creators. An attacker can't forge a transaction unless they steal a private key.
- A block structure is predefined. Attackers can't modify it to suit their purposes.
- The chain part of the blockchain is achieved using hash functions. Each block includes the hash of the previous block, creating a clear link between each block in the blockchain.
- Each block is digitally signed by its creator. The creator is selected through the blockchain's consensus protocol, making it difficult for an attacker to be a legitimate creator.

All four of these features help to make the blockchain resistant against changes occurring after the fact.

Why Is the Blockchain Immutable?

Each transaction cannot be forged or modified because it is mathematically infeasible to forge a digital signature. The structure of blocks is publicly defined, and invalid blocks will be publicly rejected.

Each block "locks in" the value of previous blocks by including their hash. Attackers cannot find another block that will produce the same hash.

A block cannot be forged or modified, because it is digitally signed by the creator. The creator of a block is either publicly known (Proof of Stake) or difficult to become (Proof of Work), making masquerading as the real creator difficult or impossible.

Now, let's take a moment to discuss how each of the features mentioned contribute to the immutability of the blockchain.

At the bottom level, each transaction is digitally signed. This means two things about transactions:

- Existing transactions can't be changed after the fact, because the signature will no longer match.
- Fake transactions can't be created since an attacker can't create a valid digital signature for a transaction between other parties.

Both of these contribute to the immutability of the blockchain since they limit the range of transactions that an attacker has to work with if he wants to create a fake but valid blockchain.

Next, the block structure is publicly defined in the protocol. This limits the types of modifications that an attacker can make to a block when trying to modify the blockchain.

Third, each block contains the hash of the previous block. This is what ties the blocks of the chain together. Remember from earlier that one of the properties of a hash function is that it is extremely difficult to find two inputs to a hash function that create the same output. Since a block contains the hash of the previous block, it's difficult to find a different version of the ledger's history that matches the most recent block, as that would require finding two different versions of the previous block that have the same hash.

Finally, each block is digitally signed by its creator. Since the creator of a block is selected via a consensus algorithm, it's difficult for an attacker to become the legitimate creator of a given block. If an attacker is not the legitimate creator of a block, it's impossible for them to create a digital signature that others would accept.

Specific Immutability Mechanisms

Now, let's look at how different blockchains implement immutability.

In Ethereum and Hyperledger, the immutability mechanism is the one that we've described previously. Each transaction and block is digitally signed and are linked using cryptographic hashes.

Corda, on the other hand, relies on its notary service for immutability. Each Corda network has one or more notary services that verify transactions. Each transaction is considered separately and, if approved, is signed by the notary service. Transactions signed by a notary are finalized and cannot be modified after the fact.

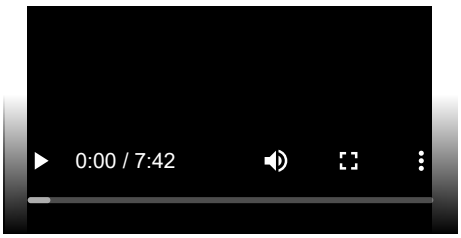
Hashing and Chaining

It is important to understand the value of locking in the previous block by including its hash in the next block.

- Creating a ledger of transactions with blocks that refer to previous blocks is a much better idea than numbering the blocks sequentially.
- In a book with pages, 1, 2, 3, etc., it would be easy to tear out page 25 and replace it with another page.
- The integrity of the book has been manipulated and altered. However, there is nothing about the new page number that ties it (chains it) to the content of the previous page.
- Instead, in a blockchain, blocks are referenced by their *hash* and each block explicitly specifies which block (hash) it is building on.

On the next page, we will take a look at how these blocks and hashes look in an actual example.

Video: Hashing and Chaining with Proof of Work



Lab 2: Blocks

Next, let's engage with an interactive lab. This lab will give you the chance to examine blocks on the blockchain. Enjoy!

Start Lab

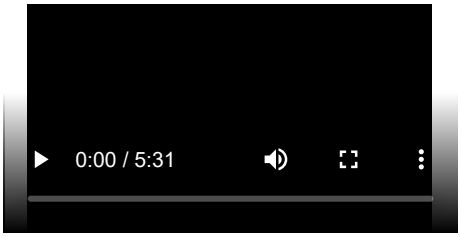
Smart Contracts

Learning Outcomes

By the end of this section, you should be able to:

- Explain what smart contracts are and how they work.
- Discuss the benefits of using smart contracts.

Video: Smart Contracts



Smart Contracts Review

Smart Contracts

- Can also be known as chaincode – program rules and decision points into blockchain transactions and processes.
- Automate transactions and ensure they are all following the same rules.
- Stored on the blockchain.
- Address limitation of the Bitcoin protocol.

What Do Smart Contracts Provide?

Blockchain Security

Learning Outcomes

By the end of this section, you should be able to:

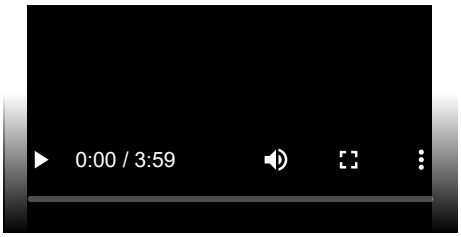
- Compare blockchain security vs. standard security.
- Compare security of public and private blockchains.

Blockchain Security vs. Standard Cybersecurity

Blockchain is commonly called the future of computing. It takes a very different approach to data storage and processing and requires a very different perspective for security.

In this section, we will discuss some of the ways that security differs in traditional and blockchain environments.

Video: Introduction to Blockchain Security vs. Standard Cybersecurity



Blockchain Security vs. Standard Cybersecurity: The Environment

One of the primary differences between cybersecurity in a traditional computing environment and on the blockchain is the environment itself and what it is and isn't designed to do.

The traditional computing environment is a company network fully or, at least mostly, under the control of the company's computer security staff. While many organizations are making the shift to cloud-based environments, they still have a high degree of control over the security and configuration of their rented systems. Traditional networks are highly centralized, and the focus of cybersecurity on these systems is primarily perimeter-focused. All systems and authorized users on the network are trusted or semi-trusted, so the focus is on preventing attackers from entering from outside the network.

Blockchain is designed to be a decentralized, distributed system running on untrusted hardware. While security in traditional environments is designed to provide security by putting all data in one place and building walls around it, security in blockchain is based on ensuring that data is protected from modification by copying data to

as many locations as possible to make modification of all copies infeasible. Traditional infrastructure focuses on confidentiality and integrity, while blockchain is designed to provide integrity and availability.

Blockchain Security vs. Standard Cybersecurity: Security

Both traditional computing environments and blockchain have security considerations associated with them. In many cases, the same attack is possible against both paradigms, but the details of how to implement it vary.

Here, we discuss how a few different attacks can be launched against traditional computing environments and blockchain:

- Denial-of-service
- Endpoint security
- Intentional misuse
- Code vulnerabilities
- Data protection.

Blockchain Security vs. Standard Cybersecurity: Denial-of-Service (DoS)

A denial-of-service (DoS) attack is when an attacker makes it impossible for a system to serve its users as designed. This can be accomplished by exploiting a flaw in the system, but, more commonly, is accomplished by performing legitimate actions at a rate higher than the target can handle.

To be effective, denial-of-service attacks typically focus on a system's weakest link or bottleneck. In traditional environments, denial-of-service attacks target a company's web server to prevent customers from accessing the company's services. This can be accomplished by making more connection requests than the server is capable of supporting. In blockchain, a denial-of-service attack involves submitting more transactions to the blockchain than it can handle.

Since many blockchains have fixed-size blocks created at a fixed rate and are stored in a distributed fashion, they have a maximum capacity that a determined attacker can exceed, rendering the blockchain unusable.

Blockchain Security vs. Standard Cybersecurity: Endpoint

Traditional infrastructure and blockchain environments also differ with regard to endpoint security. In traditional cyber, endpoints are under the control of the enterprise and have some level of heterogeneity. In blockchain, endpoints are the nodes and may be completely homogeneous.

Heterogeneity can be dangerous because an attacker has more options for finding a vulnerability to exploit, while homogeneity means that a flaw in one system is a flaw in all of the systems.

Blockchain Security vs. Standard Cybersecurity: Code Vulnerabilities

Another way that traditional cybersecurity and blockchain differ is in the level of trust in the code used in a company's applications. In traditional cyber, the company writes most of the code, and vulnerabilities can arise only from code that the company controls.

In blockchain, anyone can write a smart contract, and a flaw in the smart contract or the underlying platform cod can have wide-reaching consequences. The only hack to date against the Bitcoin network was enabled by an integer overflow vulnerability in the Bitcoin protocol.

When exploited, an attacker was able to assign himself more Bitcoin than was ever intended to be created. If the Bitcoin network didn't "break the rules" by modifying the historical ledger through a hard fork, Bitcoin would have become worthless. Anyone who wants to use Bitcoin has to accept the risks of hacks like this; they can't modify the code before including it in their application.

Blockchain Security vs. Standard Cybersecurity: Intentional Misuse

Both traditional and blockchain environments are vulnerable to attacks based on intentional misuse of the system. In traditional cyber, insider attacks or intentional misuse of the system by clients are possible. In fact, a denial-of-service attack is a specific type of intentional misuse.

In blockchain, systems using Proof of Work incentivize miners to do something a lot, but not too much. The main weakness of Proof of Work is that a blockchain becomes insecure if more than half of the mining network's processing power is controlled by a single group.

Proof of Work incentivizes miners to control as much processing power as possible to win rewards, but doesn't want them to become too successful.

Blockchain Security vs. Standard Cybersecurity: Data Protection

Finally, traditional infrastructure and blockchain differ in their goals regarding data protection. In traditional cyber, data is siloed, and access is strictly controlled by the owners, placing responsibility for confidentiality, integrity, and availability in their hands.

In blockchain, data is distributed, and the blockchain is relied upon to provide integrity and availability.

Security: Public vs. Private Blockchains

Let's discuss the differences between a **public** and a **private blockchain**:

As the blockchain continues to evolve, the terminology has become confusing. Both public and private blockchains share many similarities:

- Both are decentralized peer-to-peer networks, each maintaining a shared append-only ledger of digitally-signed transactions.
- Both maintain transaction replicas in-sync through a protocol referred to as consensus.
- Both provide certain guarantees on the immutability of the ledger.

More importantly, the main difference between a public and private blockchain is related to who is allowed to participate in the network, execute the *consensus* protocol, and maintain the shared ledger.

A public blockchain network is completely open and anyone can join and participate in the network.

A private blockchain network requires an invitation, and must be validated by either the network starter or by a set of rules. Private blockchains are usually set up as *permissioned* networks, placing restrictions on who is allowed to participate in the network, and only in certain transactions.

Lab 3: The Blockchain

Next, let's engage with an interactive lab. In this lab, we will be examining a blockchain containing 5 blocks, each labeled according to position. Enjoy!

Start Lab

Public and Permissioned Blockchains

Learning Outcomes

By the end of this section, you should be able to:

- Indicate the differences between private and public blockchains.
- Discuss about the advantages and disadvantages of public and private blockchains.
- Understand when to use a public vs. a private blockchain.

Understanding the Difference

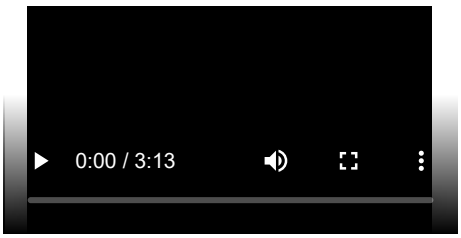
When we try to understand the main difference between a public and private blockchain, it is important to appreciate that the terminology in the media gets routinely improperly stated.

If you have some time and would like a much deeper dive into public blockchain vs. private blockchain for the enterprise, there is a great webinar on this topic: "*Public Blockchain vs Private Blockchain for the Enterprise*".

A public blockchain is really a **permissionless** blockchain. Anyone can effectively join the blockchain, meaning that they can read, write, or participate with a public blockchain. Public chains are decentralized, no one entity has control over the network, and they are secure in that the data can't be changed once validated on the blockchain.

A private blockchain is really a **permissioned** blockchain. Permissioned networks place restrictions on who is allowed to participate in the network and in what transactions.

Video: Public (Permissionless) Blockchains

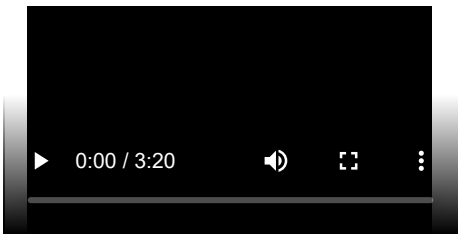


Public Blockchain Benefits

The benefits of public blockchain are:

- **Open Read and Write**
  Anyone can participate by submitting transactions to the blockchain, such as Ethereum or Bitcoin; transactions can be viewed on the blockchain explorer.
- **Ledger Is Distributed**
  The database is not centralized like in a client- server approach, and all nodes in the blockchain participate in the transaction validation.
- **Immutable**
  When something is written to the blockchain, it can not be changed.
- **Secure Due to Mining (51% rule)**
  For example, with Bitcoin, obtaining a majority of network power could potentially enable massive double spending, and the ability to prevent transaction confirmations, among other potentially nefarious acts.

Video: Private (Permissioned) Blockchains



Private (Permissioned) Blockchains

Lets discuss what private blockchains are and why they are utilized by enterprises:

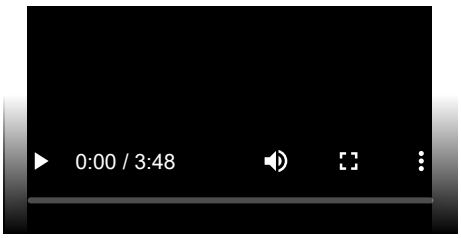
- Private blockchains are also referred to as permissioned or enterprise blockchains. Enterprises need to ensure some level of security, privacy, compliance, performance, and many of the properties that a private blockchain can provide.
- Can be open sourced, consortium, or privately developed. There are many options for a private blockchain, and the most common ones are R3 Corda, Hyperledger, and Quorum.
- Transactions are processed by select nodes in the blockchain. From a performance perspective, this is where having only a few nodes process transactions vs. 14,000 nodes in Ethereum's case can really create a performance gain around latency and transaction speed.
- Transactions are not publicly viewable (transparent) in the blockchain, and only select nodes can access the ledger.
- Locally distributed, examples include: R3 Corda can transact between nodes, and the rest of the blockchain does not participate.

Private (Permissioned) Blockchain Benefits

The benefits of private blockchain are:

- **Enterprise Permissioned**
  The enterprise controls the resources and access to the blockchain, hence private and/or permissioned.
- **Faster Transactions**
  When you distribute the nodes locally, but also have much less nodes to participate in the ledger, the performance is faster.
- **Better Scalability**
  Being able to add nodes and services on demand can provide a great advantage to the enterprise.
- **Compliance Support**
  As an enterprise, you likely would have compliance requirements to adhere to, and having control of your infrastructure would enable this requirement more seamlessly.
- **Consensus More Efficient (less nodes)**
  Enterprise or private blockchains have less nodes and usually have a different consensus algorithm, such as BFT vs. POW.

Video: Public and Private Comparison



Public and Private Blockchain Decisions

When it comes to decision making around what blockchain model to use, it's important to understand the considerations:

Blockchain Flow

Learning Outcomes

By the end of this section, you should be able to:

- Discuss about the actors in a blockchain.
- Review the history of the blockchain.
- Analyze the flow of a transaction in blockchain.

Blockchain Prehistory to Early History: Architecting

Blockchain architect designs and builds blockchain network:

- Optionally, traditional databases are set up to store data off-chain.
- Optionally, external processors are set up to allow blockchain to offload computation if necessary.
- Optionally, peer relationships are set up with external systems via system integration.

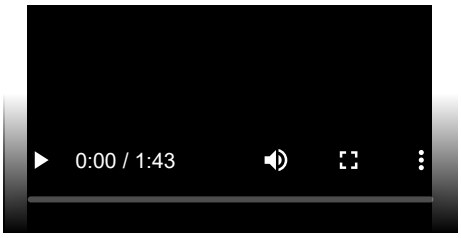
Blockchain Prehistory to Early History: Operators

Blockchain Prehistory to Early History: Users

Blockchain Prehistory to Early History: Developers and Smart Contracts

Subscribe

Video: Blockchain Transaction Flow



Blockchain Transaction Flow: Operation or Transaction Performed

Once a blockchain solution is completely set up, end users can interact with its smart contracts to take advantage of its available functionality.

Blockchain Transaction Flow: Smart Contract Triggered

To begin, a blockchain user performs an operation that should be stored on the blockchain. This can be accomplished by interacting with software that interfaces with a smart contract on the blockchain.

Blockchain Transaction Flow: Operators Spread Transaction

Once the smart contract is triggered, the relevant code is encapsulated in a transaction, which spreads to all blockchain operators through peer-to-peer transactions on their network.

Blockchain Transaction Flow: Collections of Previous and Creation of New

Through the blockchain's consensus mechanism, one of the blockchain operators is selected to be the creator of the next block on the blockchain. This operator collects all of the transactions created since the previous block into a new block, and finalizes the new block.

Annotate

Blockchain Transaction Flow: Spread New Block

This block is spread throughout the peer network through the same peer-to-peer communications as transactions.

Blockchain Transaction Flow: Execute Code

When block operators receive a copy of the new block, they add it to their copy of the distributed ledger and execute the smart contract code included in each transaction in the block. This guarantees that all members of the peer network agree on the current state of the blockchain's distributed computer.

Blockchain Transaction Flow: Operation Complete

The user's wallet monitors for the creation of new blocks that include transactions associated with the user. When a block containing the completed code from the user's operation is received, an event is created to notify the user that the operation is complete.

Consensus and Fault Tolerance

Learning Outcomes

By the end of this section, you should be able to:

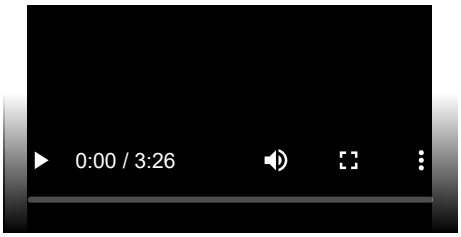
- Explain what consensus is and why consensus is needed in a blockchain.
- Explore different methods of achieving consensus (Proof of Work, Proof of Stake, etc.).
- Discuss the advantages and disadvantages of each method.
- Discuss consensus incentives.
- Analyze what consensus achieves and how.

Consensus in Blockchain

The blockchain is a distributed and decentralized system, which means that it needs to have a way of tracking the official current state of the system. Since the blockchain can include financial transactions and business agreements, it is important that all parties involved are in sync regarding the terms of the agreement.

In this section, we will discuss the details of how a blockchain network comes to agreement on the contents of the blockchain.

Video: Consensus



Introduction to Consensus in the Blockchain

The blockchain is designed to be a shared, synchronized historical ledger, meaning that there needs to be a final decision at some point on what should and shouldn't be included in the official record. Since blockchain is decentralized, there is no "higher authority" that can rubber-stamp and finalize the contents of a blockchain block.

Introduction to Consensus in the Blockchain: Scarcity

The method that Satoshi Nakamoto, the creator of blockchain, invented to achieve consensus is based on scarcity. In one way or another, blockchain consensus algorithms boil down to some kind of vote where the number of votes that a user has is tied to the amount of a limited resource that is under the user's control. Based on the economic Laws of Supply and Demand, collecting enough of an asset to have a controlling share will drive up the price of the asset enough to make achieving that level of control unfeasibly expensive.

Introduction to Consensus in the Blockchain: Consensus Mechanisms

Satoshi Nakamoto invented a consensus algorithm called Proof of Work for the use of Bitcoin. Since then, several other consensus algorithms have been invented to fit different use cases. These include Proof of Stake, Delegated Proof of Stake, Practical Byzantine Fault Tolerance, and Directed Acyclic Graphs. The most commonly used consensus algorithms are Proof of Work and Proof of Stake.

Proof of Work: Computational Resources

Proof of Work: Incentivizes

In Proof of Work, users in the blockchain network who want to create the next block (and win the associated reward) are called miners. To win the right to mine a block, miners race to find an acceptable solution to a “hard” cryptographic problem. As we discussed previously, “hard” mathematical problems can only be solved by random guessing. When a miner finds an acceptable solution, they create a block and broadcast it to the network, finalizing that block.

Proof of Work exploits the scarcity of computational resources by choosing a problem that can only be solved by guessing. There is no limit on the number of guesses that a miner can make at once. Proof of Work, therefore, incentivizes miners to run as many mining machines as possible to maximize the probability that they are the first to find a solution to the problem. Since mining computers take money to purchase and money to run, the amount of control that a user can exert over the blockchain is limited by the amount of money they have available to invest in mining equipment.

Proof of Work: 51% Security

The security of the Proof of Work consensus is based on the assumption that no one controls more than half of the computational resources of a blockchain's mining network. If this was the case, the miner has a high probability of finding an acceptable solution to the mining puzzle before anyone else for every block in the blockchain. This gives the miner complete control of the blockchain and breaks the decentralization of blockchain.

Proof of Stake: Scarce Commodity

Proof of Stake: Stake

Users in a Proof of Stake blockchain can “stake” or promise not to use the tokens they own. This gives them the opportunity to be selected as the next user to create or “forge” a new block and earn the reward. A block forger is pseudo-randomly selected from all of the users who have staked some of their assets, and the selection process is biased based on the size of the stake.

For example, imagine that a wheel is divided into sections where the size of a section is proportional to the size of a user’s stake. The next block creator would be chosen by spinning the wheel and seeing whose section comes out

on top. In Proof of Stake, each user has a copy of the wheel and they are all synchronized so that each person can independently determine the selection and get the same result. This is why Proof of Stake uses a pseudo-random instead of a random selection process.

Proof of Stake: Economic Infeasibility

In Proof of Stake, an attacker needs to control enough of the staked currency to guarantee they will be selected to create every block. Since cryptocurrency is a limited asset, buying up enough of it to do this is expensive, making attacks on Proof of Stake systems economically infeasible.

Specific Consensus Implementations: Ethereum

Ethereum currently uses Proof of Work for consensus. And Casper is the planned migration of Ethereum from Proof of Work to Proof of Stake.

Of the three blockchains studied, Ethereum is the only one that uses a standardized consensus mechanism. Ethereum was designed from the beginning to use Proof of Work for consensus, until a forced hard fork to the Proof of Stake implementation (codenamed Casper). This forced hard fork is baked into the Ethereum protocol and will be accomplished by slowly increasing the difficulty of the Proof of Work problem until the time taken to solve it increases to the point where Proof of Work becomes unusable. Proof of Stake does not require the same energy consumption as Proof of Work and is a more sustainable and scalable consensus mechanism.

Specific Consensus Implementations: Hyperledger Fabric

Hyperledger Fabric breaks out consensus into components, allowing users to pick a consensus algorithm for their particular use.

Hyperledger Fabric deliberately avoided hard-coding a consensus mechanism into the protocol by defining an "orderer component" that performs all of the consensus-related operations. This allows users of Hyperledger Fabric to select a consensus algorithm that fits their use case without being forced to make large-scale code edits.

Specific Consensus Implementations: Corda

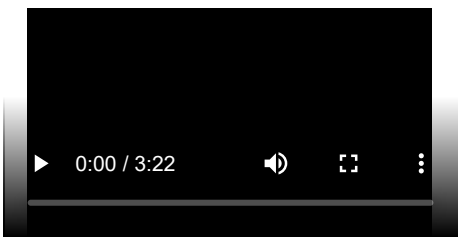
Each **Corda** network has a notary service made up of independent parties that approve blocks using any applicable consensus algorithms.

Corda does not follow the standard blockchain model of transactions being bundled into blocks and then being finalized by the network as a whole. Instead, a Corda network contains one or more notaries consisting of several independent parties. Transactions in Corda are finalized by a notary with a multiparty digital signature using an algorithm like Raft.

Fault Tolerance in the Blockchain

Blockchain is a distributed, decentralized system that maintains a shared state. While consensus algorithms are designed to make it possible for the network to agree on the state, there is the possibility that agreement does not occur. Fault tolerance is an important aspect of blockchain technology.

Video: Fault Tolerance



The Byzantine Generals' Problem

The **Byzantine Generals' Problem** (discussed in the previous video):

- Several generals needing to agree on a coordinated plan of attack.
- One or more generals may be traitors.
- All generals will abide by the majority decision, but may try to influence it.

Blockchains are designed to have Byzantine Fault Tolerance:

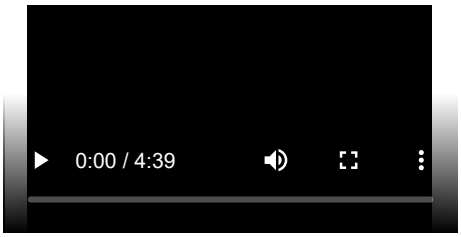
- All nodes are untrusted.
- Nodes must come to a consensus on the official state of the blockchain.

The Byzantine Generals' Problem is a scenario designed to demonstrate the difficulty of multiple parties coming to an agreement when communication can only be accomplished on a one-to-one basis and is untrusted. In the story, several Byzantine Generals are besieging a city with their separate armies. If they all attack together or all retreat together, they will be ok, but if some attack while others retreat, they will be destroyed.

The generals can only communicate by messengers, who could be intercepted and forced to carry fake messages, and one or more generals may be a traitor. The goal is to find a way to achieve a consensus on strategy despite the possibility of traitors and false messages. Presumably, all generals will abide by what they believe is the majority consensus. The Byzantine Generals' Problem is solvable as long as two-thirds of the generals are honest.

Blockchain is designed to be Byzantine Fault Tolerant, meaning that the network will come to a consensus on the official state of the blockchain, despite the fact that some members may misbehave. The solution to the Byzantine Generals' Problem is inefficient, so the blockchain needs some way of being confident of consensus without going through a full solution.

Video: Proof of Work vs. Proof of Stake



Proof of Work vs. Proof of Stake

Proof of Work	Proof of Stake
Distributed consensus among untrusted and unidentifiable nodes	Distributed consensus among untrusted and identifiable nodes
Incentives are reworded within the system for work done outside of the system	Incentives are rewarded within the system for escrow inside the system
Relatively high cost of entry, but high returns	Low cost of entry, but low returns
Empirically proven	Experimental

Proof of Work provides a game-theoretical distributed consensus algorithm:

- Proof of Work incentivizes mining nodes on the network to reach for the thermodynamic limit of computational cycles. This incentivizes decentralization because heat from mining nodes dissipates better in two separate places rather than one centralized location. Note, this decentralization is solely physical and a network distribution.
- Proof of Work has empirically proven that game-theory can be weaved into a protocol because it successfully applies incentives at every possible action within the network.
- Proof of Work only works because it is optimization-free and approximation-free.
    1. Optimization-free means there is no possible way to circumvent the hashing of the mining protocol necessary to secure a block.

2. Approximation-free means there is no possible way to almost have a block. The process is binary; there are blocks and not blocks.

Proof of Stake provides an experimental internally game-theoretical consensus algorithm:

- It relies on nodes already having cryptocurrency to stake. It rewards nodes with the most money staked, and not the most computational power.
- It requires that each validating node be identifiable. This is because the staked coins must be held accountable for any malicious acts. Proof of Work does not require identification.
- In Proof of Stake,you are competing with a much larger group of nodes. There is no transactional friction involved in staking coins, unlike in Proof of Work, which requires buying mining hardware, hooking up internet, providing cooling systems, etc.

Governance and Blockchain

Learning Outcomes

By the end of this section, you should be able to:

- Explain what governance is.
- Discuss the different types of governance and what types of governance are used in the blockchain.

What Is Blockchain Governance?

All organizations and software development projects need a way to finalize each decision along the roadmap. Most organizations are centralized and have a set leadership team. Several strategies for governing the decentralized blockchain have been developed.

Effective blockchain governance includes:

- Incentives
- Methods of coordination.

Before getting into the details of how governance works on the blockchain, it's important to have a clear definition of what blockchain governance is. Every blockchain is an evolving system that needs to change to meet the needs of its users. If a blockchain isn't relevant and useful, then it won't survive.

To evolve, the blockchain needs to make changes and needs a way to make final decisions on what these changes should be. Most organizations have a leadership team or a CEO who is the final authority for their organization. However, blockchain is designed to be decentralized, and not be under the control of any person or group. This means that blockchain needs another way to make decisions regarding the blockchain's roadmap.

For blockchain governance to be effective, it needs to include both incentives and methods for members to coordinate. Without incentives, members won't participate in governance and the blockchain will become less aligned with user needs over time. Without a method for members to coordinate, it will be impossible for a blockchain network to come to an agreement on future changes.

Blockchain Governance Strategies

Several different blockchain governance strategies have been proposed and implemented for different blockchains. Here, we will review some blockchain governance strategies sorted from the fewest to the most members directly involved in the decision:

- "Benevolent Dictator for Life"
- Core Development Team
- Open Governance
- On-Chain Governance.

Blockchain Governance Strategies: Benevolent Dictator for Life

The original creator or lead developer of a cryptocurrency has the final say on all decisions.

The simplest blockchain governance strategy is nicknamed "**Benevolent Dictator for Life**". In this strategy, the creator of the blockchain is the final authority on all decisions regarding the blockchain. A good example of this type of leadership is Facebook, where Mark Zuckerberg has the final say on the future roadmap of the Facebook platform.

Blockchain Governance Strategies: Core Development Team

A team of the most active developers decides what functionality should or shouldn't be included.

The next step up places control of the blockchain roadmap in the hands of a **Core Development Team**. This is a strategy commonly used in open source programming projects, where users are able to offer or request features, but developers have the final say on what is or is not included in the official release.

Blockchain Governance Strategies: Open Governance

The team making governance decisions for the blockchain is chosen by the users of the blockchain.

Some blockchains use the **Open Governance** method of handling governance of the blockchain. In this system, the team that makes the final technical decisions for a system is selected by the system's users.

Blockchain Governance Strategies: On-Chain Governance

The rules for how the blockchain operates are stored on-chain in smart contracts with built-in capability and procedures for modifications.

A blockchain-specific governance strategy is **On-Chain Governance**. In this form of blockchain governance, the rules describing how the blockchain should operate are stored on the blockchain itself. These regulations typical are implemented as smart contracts on the blockchain with built-in methods for users to modify the rules based upon their needs and the needs of the blockchain.

Who Really Governs the Blockchain?

Blockchain governance comes down to the users.

Major changes to a blockchain require a hard fork. A hard fork is a change to the blockchain protocol that makes it incompatible with old clients.

- For a hard fork to be successful, users need to agree to follow it.
- Users can refuse to follow a hard fork, creating a divergent blockchain. The DAO Hard Fork on Ethereum created Ethereum Classic.

Despite the official story of who governs the blockchain, in the end, the users are the ones who really make the final decisions of what will or will not be included in the blockchain. With the huge number of potential options, users can abandon a blockchain that makes changes that they disagree with.

Any major change to a blockchain requires a “hard fork”. All this means is that the blockchain protocol has changes that are not backward compatible, so blockchain clients that do not make the switch will not be able to operate on the main blockchain. For a hard fork to be successful, users of the blockchain need to make the decision to update their clients to incorporate the new changes.

If not all users decide to make the switch after a hard fork, a divergent blockchain can be created. Since the blockchain is a distributed network, the decision to implement a hard fork doesn’t cause the old version of the blockchain to become non-functional. Users who choose not to follow the fork can decide to maintain the old blockchain, fragmenting the blockchain network.

DAO Hard Fork on Ethereum Created Ethereum Classic

One famous example of this type of fragmentation is the DAO hack on the Ethereum network. The DAO was an Ethereum smart contract that completed a record-breaking crowdfunding campaign on the Ethereum network, with all of this value stored within the DAO smart contract. A flaw in the smart contract's code allowed an attacker to create another version of the smart contract under their control and siphon off a portion of the DAO contract's funds, worth roughly 72 million dollars at the time. After much debate, the Ethereum network decided to implement a hard fork that allowed investors of the DAO to reclaim their stolen Ether.

This was a very contentious decision because the historical ledger in the blockchain is supposed to be immutable and all transactions are final. Smart contracts are supposed to be their own final authority, so any action that could be performed with a smart contract, including exploiting a programming flaw to drain value from it, is considered fair game. The Ethereum network's decision to reverse the DAO hack went against the principles of blockchain's immutability and the supposed self-regulation of smart contracts.

Some of the Ethereum network refused to follow the DAO hard fork, resulting in a divergent blockchain where the DAO hack was successful. This created the Ethereum Classic cryptocurrency, which shares the same history as Ethereum up to the DAO hack, but is completely independent after that point. Despite the "official" decision to reverse the DAO hack, users made the final call of whether or not they would abide by that decision.

Governance in Ethereum

We've been discussing the decisions made regarding the Ethereum blockchain, but haven't discussed who made those decisions yet. Ethereum uses the "Benevolent Dictator for Life" mode of blockchain governance. While user input and input from the development team is welcome for Ethereum, Vitalik Buterin is the final authority on decisions regarding the Ethereum roadmap.

Governance in Hyperledger Frameworks

Hyperledger frameworks, on the other hand, use an Open Governance model to make technical decisions regarding the Hyperledger environment. The Hyperledger Technical Steering Committee (TSC) is the final authority for technical decisions in Hyperledger.

Each year, the Hyperledger Technical Steering Committee is selected from the Hyperledger environment's active contributors and maintainers.

Contributors and maintainers can submit themselves as potential candidates for the eleven slots, and the slots are filled based on voting by the same group of contributors and maintainers. This model is designed to allow those with an active role in the Hyperledger development community to have a say in how that community is governed.

Governance in Corda

Corda also uses an Open Governance model to make technical decisions regarding the future of the blockchain. The Corda Network Governing Body will be selected to represent the interests of all users in the Corda network.

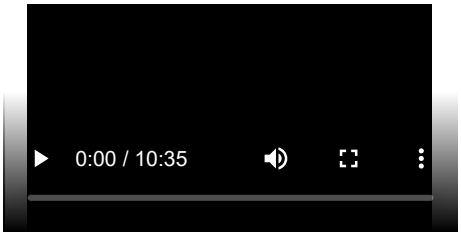
Identity and Anonymity on Blockchain

Learning Outcomes

By the end of this section, you should be able to:

- Understand why anonymity is required in public blockchains.
- Discuss how private/public key cryptography provides anonymity.

Video: Identity on the Blockchain



Identity: Public Key Cryptography

Identity in the blockchain is based on public key cryptography. A person's address on the blockchain is their public key.

Transactions on the blockchain include their public key and are digitally signed with the sender's private key:

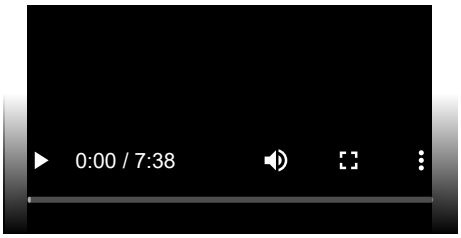
- A digital signature verifies that someone in possession of the private key authorized the transaction.
- Digital signatures can be easily verified using the corresponding public key, which is included in the transaction.

Identity: RSA Public Key Cryptography

Identity: Specific Identity Implementations

- **Ethereum**
  A user's identity is an address based on their public key.
- **Hyperledger**
  Identity is managed by X.509 certificates.
- **Corda**
  Identity is managed by X.509 certificates:
  1. Public: Certificates published on blockchain
  2. Confidential: Certificates only shared with parties involved in transaction.

Video: Anonymity in the Blockchain



Let's Review

Public key cryptography gives each person a pair of keys that work together:

- **Private keys**
  They can decrypt things that public keys encrypted. They can only be tied to an identity if the owner wishes.
- **Public keys**
  They can verify signatures made with private keys. They can only be tied to a private key if the owner wishes.

Advanced Blockchain Anonymity Techniques

The following are only some of the mechanisms developed and implemented in various blockchains:

- **Zero-Knowledge Proofs**
  A prover proves knowledge of a secret without revealing it.
- **Stealth Addresses**
  Using one-time addresses for sending/receiving transactions for an account.
- **Ring Signatures**
  Type of digital signatures that lets any member of the group sign, but no one can tell which one signed.
- **CoinJoin**
  Transactions from several senders to several recipients are mixed together to hide who is paying whom.
- **Confidential Transactions**
  Uses homomorphic encryption to allow transactions to be processed while encrypted. Proves transaction value is in a range of values to prove that overspending did not occur.

Advanced Blockchain Anonymity Techniques: Zero-Knowledge Proofs

Zero-knowledge proofs use cryptographic algorithms to allow a user to prove knowledge of a secret without revealing the secret.

Advanced Blockchain Anonymity Techniques: Stealth Addresses

Stealth addresses involve using one-time addresses to perform transactions on a blockchain. A stealth address is just a one-time address that makes it impossible to link a transaction to a known account. This prevents the data

mining attacks on privacy that we discussed earlier.

Advanced Blockchain Anonymity Techniques: Ring Signatures

We mentioned previously that transactions are digitally signed. With ring signatures, all that can be determined from a transaction is that a member of a group signed it, but not the particular member.

Advanced Blockchain Anonymity Techniques: CoinJoin

The ability to see who is performing transactions with whom is dangerous to user privacy and anonymity. Protocols like CoinJoin mix several transactions together so that it is difficult to pair senders with recipients.

Advanced Blockchain Anonymity Techniques: Confidential Transactions

Confidential transactions take advantage of homomorphic encryption, which makes it possible to perform mathematical operations on encrypted data. This means that the data contained in a transaction can be hidden from the public, while still allowing the network to verify that the transaction is valid.

Specific Anonymity Implementations

- **Ethereum**
  Ethereum currently does not have any advanced privacy options, but this is planned to change.
- **Hyperledger**

    1. Channels: Subsections of the blockchain that make transactions visible only to members.
    2. Private Transactions: Hashes of private data are stored to publicly verify it on the blockchain.
    3. Zero-Knowledge Technology: Provers can demonstrate knowledge of a secret without revealing the secret itself.
- **Corda**
  Parties on the Corda Network can be represented in one of two ways:

    1. Party: A public key and name
    2. Anonymous Party: Only a public key.

Trust and Trustless

Learning Outcomes

By the end of this section, you should be able to:

- Explain how a blockchain is “trustless”.
- Explain why a trustless system is more secure than a system that requires trust.

Trust in Blockchains

Just as there are benefits with blockchain technology, there are also some challenges. Blockchain is a culmination of technologies that have been blended to provide a trustless platform. Expect some challenges and use case justifications taking the old line of business apps to the blockchain.

Let’s recap the features of a blockchain that establishes trust:

- Blockchain technology is about storing some kind of data (which are transactions in regards to the Bitcoin blockchain).
- Blockchain is essentially transferring trust from an intermediary to technology.
- Storing data in the blockchain is through cryptographic functions.
- Private key/public key.
- Collaboration through consensus.

Establishing Trust in Blockchains

All transaction data on a chained block is assumed to be trustworthy.

The users base this trust on the fact that:

- This data has not been tampered with
- The blockchain is immutable.

What Do Blockchains Really Do?

Blockchains minimize the amount of trust required from any single actor in the system. They do this by distributing trust among different actors in the blockchain as defined by the consensus protocols.

Blockchains have a shared ledger that gives us the absolute truth of the state of the system. It uses mathematics, economics, and game theory to incentivize all parties in the system to reach a “consensus” (i.e. coming to an agreement on a single state of the ledger).

Trustless Blockchains

Summary

Chapter 1: Summary

Blockchain is a digital decentralized ledger.

Blockchains are important because they provide a safe and secure way for people to make any type of transactio without having to trust anyone.

Blocks in a blockchain can be thought of as a sheet of paper. Blocks, just like paper, can hold any type of data on them.

When blocks fill up with data, transactions are hashed into what is known as a Merkle tree. Merkle trees provide for an easy way to find any specific transaction in a blockchain.

A hash function is a one way function that takes any type of data and converts it into a unique character code. Merkle trees use hashing to convert every transaction in a block into a 20-digit character code known as the Merkle root. Hashes are also useful when comparing large amounts of data.

A block header is a hash of many things determined by the blockchain, but most frequently consists of the previous block header hash, the Merkle root of the current block, and the timestamp.

By including the previous block's header hash, blocks are "chained" together.

Chaining is important because blockchains are kept on millions of nodes across the network.

Chaining allows blockchains to easily check and see if any data was altered just by comparing the hash of the current header. If the hash is the same on every node, then the blockchain is the same. If the header hash is different in any way, then the different hash's blockchain is updated to match the majority of blockchains. This is what makes blockchains fault tolerant and immutable.

Blockchains are fault tolerant because if any one node loses track, it will be updated to match the majority of nodes running the current blockchain.

Blockchains are immutable because the data on a block can never be changed or deleted.

Contrary to a traditional database, every transaction on a blockchain is made public, and everyone can write onto a blockchain. This requires users to be anonymous to avoid identities being tied to a specific transaction.

Anonymity is achieved through public key/private key cryptography. Your private key is for your eyes only. Your public key can be shared with the public. Your public key is the address you receive and send transactions from. To prove that your public key is associated with your private key, a digital signature is used. A digital signature uses math to show a relation to your public key from your private key, without revealing your private key.

Anonymity poses a problem when it comes to trust. How can we be 100 percent certain that anonymous users are being honest when adding transactions to a ledger that once added, cannot be changed or deleted. The answer is to validate every transaction before adding them to the chain. This problem of validation is often referred to as the byzantine general's problem, and the solution is found with consensus algorithms like Proof of Work and Proof of Stake. These consensus algorithms take advantage of the fact that the majority of users on a blockchain have a common interest to keep the blockchain honest. We will go further into some consensus methods in chapter 2.

Not all blockchains use anonymity however. Private blockchains allow for the use of permissions to control who can read and write onto a blockchain. Private blockchains often require trust, but are much more efficient due to the lack of need for a consensus algorithm like Proof of Stake. A private blockchain would be useful when you want an extra layer of transparency and higher level of security than a traditional database might be able to offer.

Blockchain: Understanding Its Uses and Implications – Chapter 4. Blockchain Use Cases

Learn more

Blockchain: Understanding Its Uses and Implications – Chapter 3. Blockchain Problem Solving

Learn more

Blockchain: Understanding Its Uses and Implications – Chapter 2. Governance and Consensus

Learn more

Blockchain: Understanding Its Uses and Implications – Chapter 1. Introduction to Blockchain

Learn more

Read More

Share with:

Join @LearnThingsOnline on Telegram

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Subscribe

Search ...

LEARNTHINGS.ONLINE TELEGRAM GROUP

Don't have Telegram yet? Try it now!

Learn Things Online
65 members, 2 online
This group build to share some materials to learn blockchain online & news. Check LearnThings.Online
View in Telegram
If you have **Telegram**, you can view and join
Learn Things Online right away.

RSS

IPFS for Beginners – Interact With IPFS By Javascript
In this article, we'll learn how to interact with IPFS by javaScript programming language. It's one way to make your own application to interact with IPFS. The post IPFS for Beginners – Interact With IPFS By Javascript appeared first on LearnThings.Online.

On May 19th, 2020, the Libra association appoint Robert Werner, an Ex-HSBC & Ex-Goldman Sachs the founder and CEO of GRH Consulting, as its general counsel. The post Libra Appoints It's General Counsel, a Former HSBC, and Goldman Sachs appeared first on LearnThings.Online.

Subscribe

Facebook Rename Its Libra Wallet Project Calibra to Novi

2020 May 26, Facebook rename its Libra wallet project Calibra to Novi. It makes its name more separate from Libra. Novi plans to launch its App in 2020. The post Facebook Rename Its Libra Wallet Project Calibra to Novi appeared first on LearnThings.Online.

Libra Appoints It's General Counsel, a Former HSBC, and Goldman Sachs