



## Blockchain Technology (BerkeleyX: CS198.2x )

This course is from edX

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

---

Developed by Blockchain at Berkeley and faculty from UC Berkeley's premier Computer Science department, this course provides a wide overview of many of the topics relating to and building upon the foundation of Bitcoin and blockchain technology.

The course covers many key topics in the blockchain space. First, we take a look at distributed systems and alternative consensus mechanisms, as well as cryptoeconomic and proof-of-stake. We then move on to the fundamental applications of bitcoin and blockchain technology, including exploring enterprise blockchain implementations (JP Morgan's Quorum, Ripple, Tendermint, and HyperLedger), the challenges and solutions around scaling blockchain adoption, and the measures that the government is taking to regulate and control

blockchain technology. We wrap up the course by also taking a look at the various blockchain ventures today and conclude with a blockchain-based future thought experiment.

This course is open to anyone with any background. Whether you are planning your next career move as a blockchain developer, crypto trader, data analyst, researcher, or consultant, or are just looking for an introduction to Blockchain. This course will help you begin to develop the critical skills needed to future-proof your career.

## Syllabus

### Distributed Systems and Alternative Consensus

Blockchain architecture is built on the foundation of decades of computer science and distributed systems literature. We start out by providing a formal definition of distributed consensus and presenting foundational theoretical computer science topics such as the CAP Theorem and the Byzantine Generals Problem. We then explore alternative consensus mechanisms to Bitcoin's Proof-of-work, including Proof-of-Stake, voting-based consensus algorithms, and federated consensus.

### Cryptoeconomics and Proof-of-Stake

We examine the meaning and properties of cryptoeconomics as it relates to its two compositional fields: cryptography and economics. We then look at the goals of cryptoeconomics with respect to distributed systems fundamentals (liveness, safety, data availability) and the griefing factors and faults in the way of these goals.

[Subscribe](#)

### Enterprise Blockchain: Real-World Applications

We look at various existing enterprise-level blockchain implementations, such as JP Morgan's Quorum, Ripple, Tendermint, and HyperLedger. We also explore business and industry use cases for blockchain, ICOs, and the increasing regulations surrounding blockchain.

### Scaling Blockchain: Cryptocurrencies for the Masses

One major obstacle to widespread blockchain adoption is the problem of scalability. We define scaling first as it relates to Bitcoin as a payment method, and compare it to more traditional forms of payment such as credit cards. We then consider the general blockchain scalability debate and look into some of the solutions that have been proposed for vertical scaling (e.g. blocksize increases, Segregated Witness, and the Lightning Network), as well as horizontal scaling (e.g. sidechains, sharding).

### Regulation and Anonymity

We look into the measures that governments have taken to regulate and control blockchain technology. We examine Anti-Money Laundering (AML) and Know Your Customer (KYC) regulations, anonymity goals, and government techniques for deanonymization of entities on

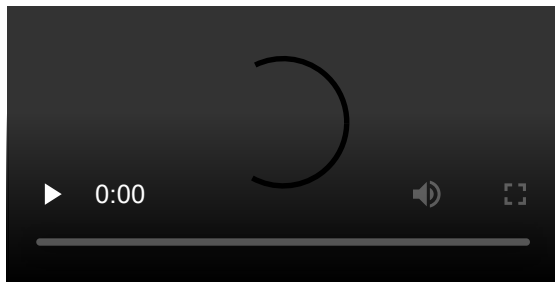
blockchain. Then from the user's perspective, we also dive into privacy oriented altcoins and mixing techniques.

## A Blockchain-Powered Future

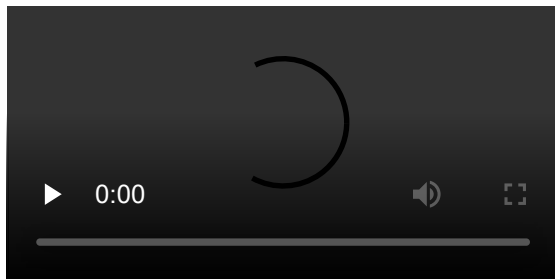
A summary of the course and an exploratory look into blockchain ventures today, such as venture capitalism, ICOs, and crowdfunding. We conclude with a blockchain-based future thought experiment.

# Trust without Trust: Distributed Systems & Consensus

## Welcome to Week 1

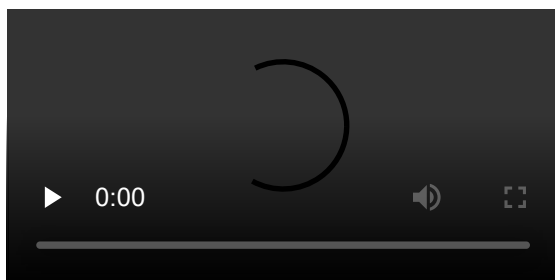


## Intro: Distributed Systems

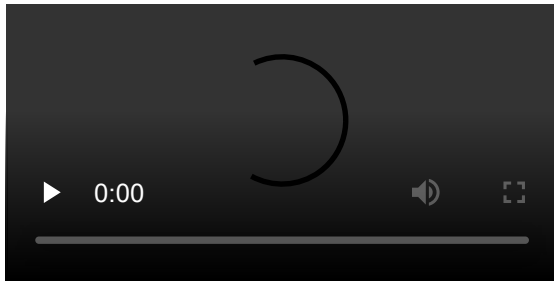


Subscribe

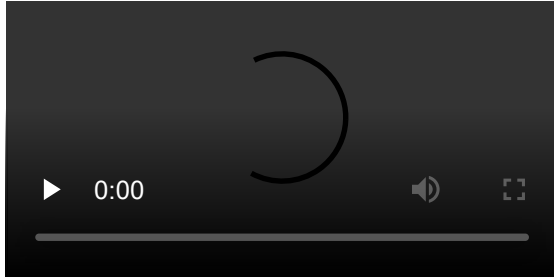
## Distributed Systems Origins



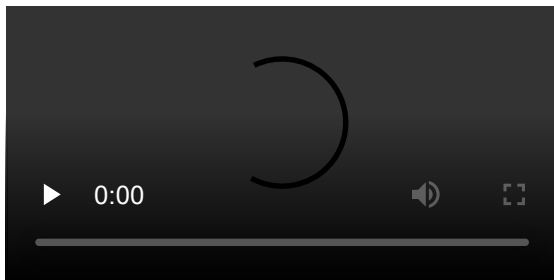
## Distributed Systems Fundamentals



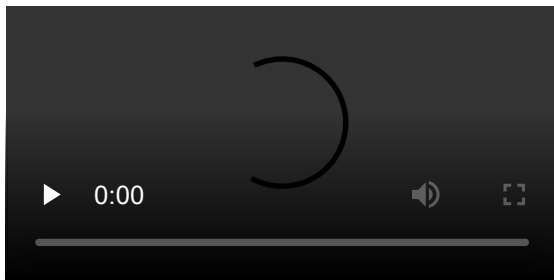
## CAP Theorem



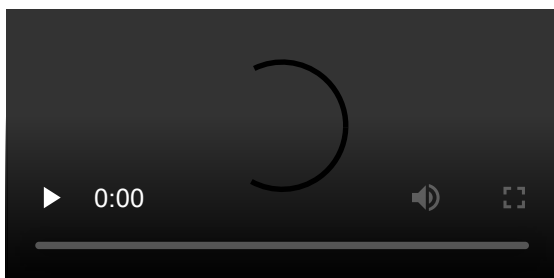
## Byzantine Fault Tolerance



## Intro: Voting Based Consensus

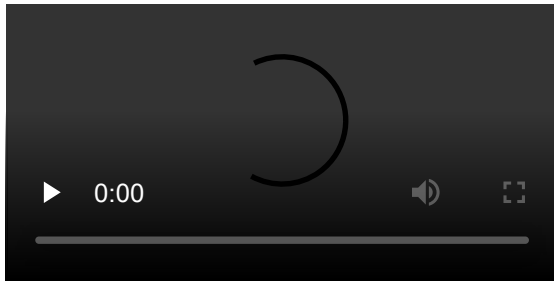


## Paxos & Raft

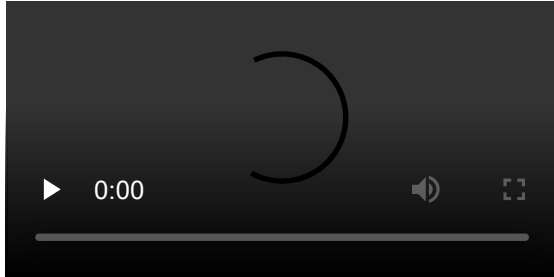


## Practical Byzantine Fault Tolerance

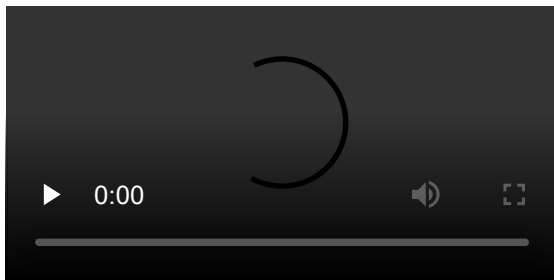
Subscribe



### Intro: Nakamoto Consensus

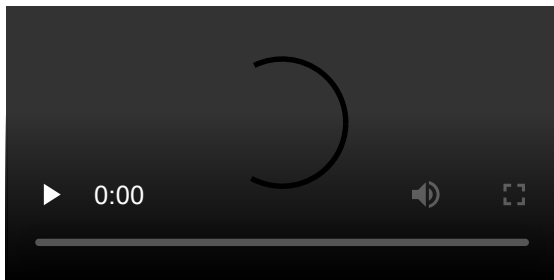


### Nakamoto Consensus

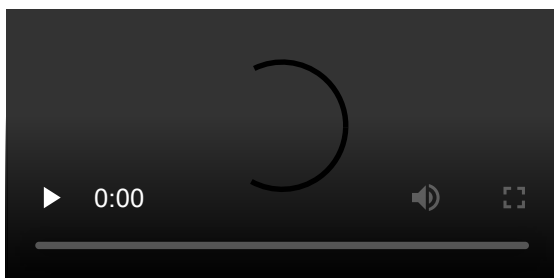


Subscribe

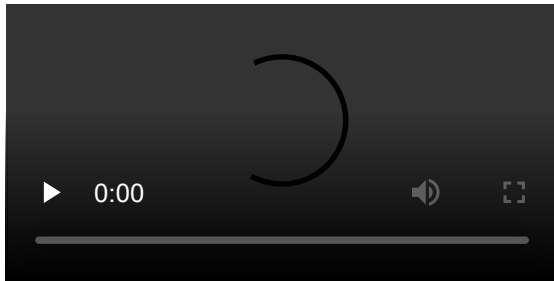
### Intro: Proof-of-Stake



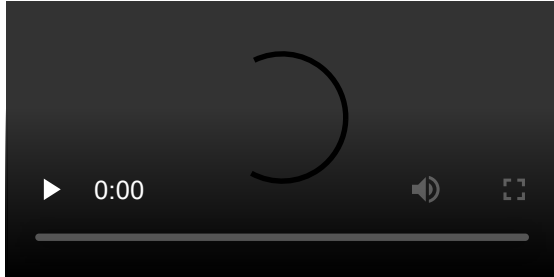
### Proof-of-Stake Overview



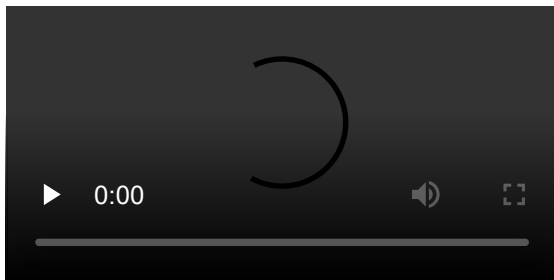
### Proof-of-Stake Implementations



### Intro: Federated Consensus

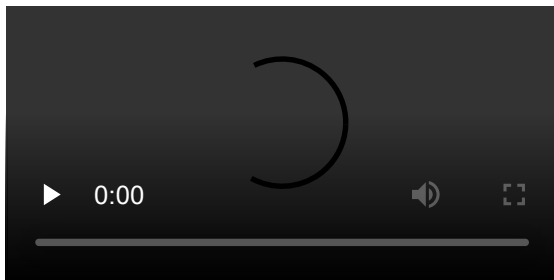


### Federated Consensus Overview



Subscribe

### Federated Consensus Implementations



### Text: Week 1 Summary

Author: Deven Navani & Nick Shen

## I. Distributed Systems

Bitcoin transactions are recorded on the blockchain, a ledger that is maintained by a **distributed system**, or a network of independent **nodes** connected by **message channels** that move information between them. A critical aspect of a distributed system is the way in which these nodes, which are unknown and untrusting of each other, come to

agreement, or consensus. In the case of Bitcoin, the network needs to agree on the number of bitcoins each individual owns, and of all transactions being made.

Distributed systems remove the need for trust in potentially unreliable parties; instead, we can trust the mathematics and the correct operation of these systems. If one of the nodes crashes or is corrupted by malicious entities, we can still protect our information and services by relying on previously set protocols that withstand these failures.

There are 3 key components of distributed systems:

- Components in the system process information **concurrently**
- Each node maintains its own clock; there is **no global clock**
- Protocols **protect against potential failure** of individual components

A distributed system is considered “**correct**” if it comes to consensus on an answer — given an input, the nodes must agree on an output.

To prove the correctness of a distributed system, we use the scheme designed by Lamport. The scheme says that a system is correct if two things are true:


- **Safety:** It doesn't do bad things!
- **Liveness:** It will eventually do good things.

To ensure correctness, we use **consensus algorithms**. There are 3 requirements of any correct consensus algorithm:

- **Validity:** any value agreed upon must be proposed by one of the processes
- **Agreement:** all non-faulty processes must agree on the same value
- **Termination:** all non-faulty nodes eventually decide

Notice that validity and agreement are safety properties while termination is a liveness property.

## II. CAP Theorem

 CAP Theorem Venn diagram highlighting intersections of Consistency and Availability with the letters CA, Consistency and Partition Tolerance with the letters CP, and Availability and Partition Tolerances with the letters AP.

The CAP Theorem states that any distributed system can only achieve 2 of the following 3 properties at any given time:

- **Consistency:** every node provides the most recent state of the system
- **Availability:** every node has constant read and write access
- **Partition Tolerance:** ability to function in spite of partitions in the network, where a **partition** is the inability for two or more nodes to communicate with each other; this is almost a given for any distributed system

Subscribe

It is important to understand there aren't black and white tradeoffs between these three properties — compromises can be made.

### III. Byzantine Fault Tolerance

**Byzantine nodes** may act maliciously or arbitrarily. Achieving consensus when  $\frac{1}{3}$  or more of the nodes are Byzantine nodes is impossible.

There are two types of **faults** that may be produced by Byzantine nodes, where faults are deviants from protocol:

- **Fail-stop:** a node can crash and not return values
- **Byzantine:** in addition to above, nodes can also send incorrect/corrupted values; all deviations from protocol fall under this category

### IV. Voting Based Consensus Mechanisms

These mechanisms allow nodes to come to consensus when no more than  $\frac{1}{3}$  of the nodes are Byzantine nodes.

**Paxos** – Consensus mechanism inspired by the Paxon parliament, who used the Paxos algorithm to pass decrees and make sure everyone on the island was in consensus. Assumes nodes do not try to subvert protocol; only works for fail-stop, no byzantine failure tolerance.

- **Proposer:** proposes legislation/changes to current state
- **Acceptor:** decides whether to accept proposed legislation
- **Learner:** learns and distributes changes to mass audience
- **Quorum:** Majority of acceptors, any two quorums must overlap

Subscribe

**Raft** – Leader based approach designed to be more understandable than Paxos, easier to implement; i.e. JP Morgan's Quorum (enterprise version of Ethereum).

- **One and only one leader:** communicates with client directly, responsible for log replication on other servers, leads until fails or disconnects
- **Leader election:** leader sends heartbeats to signal it is online and functioning; if no heartbeats are received the first node to realize a lack of leader becomes the new leader

**Practical Byzantine Fault Tolerance** – fast, handles  $F$  faults in a system with  $3F + 1$  nodes, BFT-NFS implementation only 3% slower than standard unreplicated NFS

Operates using 3 phases:

- **Pre-prepare:** the primary node sends out pre-prepare messages to everyone in the network; a node accepts the pre-prepare message so long as its valid.
- **Prepare:** If a node accepts a pre-prepare message, it follows up by sending out a prepare message to everyone else; prepare messages are accepted by receiving nodes



so long as they're valid, again, based on sequence number, signature, and other metadata; A node is considered "prepared" if it has seen the original request from the primary node, has pre-prepared, and has seen  $2F$  prepare messages that match its pre-prepare – making for  $2F + 1$  prepares.

- **Commit:** If a node receives  $F + 1$  valid commit messages, then they carry out the client request and then finally send out the reply to the client. The client waits for  $F + 1$  of the same reply. Since we allow for at most  $F$  faults, we need to wait for at least  $F + 1$ , and this ensures the response to be valid.

## V. Nakamoto Consensus

Used in Bitcoin and other cryptocurrencies. Whereas the voting based consensus mechanisms covered above use explicit voting, Nakamoto consensus uses **implicit voting** i.e. voting based on lottery-selection and earned voting power.

Nakamoto consensus is **very robust**:

- Anyone can join or leave the network at any time
- Anyone can even send corrupted messages to others
- Any user can have as many virtual identities/key pairs, as they want
- To prevent unfair voting from anyone who dishonestly creates multiple identities, voting power must be made scarce, done by tying voting power to a scarce resource such as power or electricity

Each Nakamoto Consensus protocol must have a set of rules defining how to choose the n valid state of the network, such as the **longest chain** policy in Bitcoin and many cryptocurrencies. This is because each node in the Nakamoto consensus network gets to choose its own state, and try to convince others of its validity.

Subscribe

Multiple forms of Nakamoto Consensus:

- **Proof of Work** – current blockchain standard, led by bitcoin and followed by most networks, led to mining craze and rapid acquisition of computing hardware
- **Proof of Stake** – experimental protocol to end electricity drain by staking tokens, can mine or validate block transactions according to how many tokens staked
- **Proof of Activity** – a proof of work and proof of stake hybrid protocol
- **Proof of Capacity/Space** – memory-hard PoW, allocating amount of memory or disk to solve a challenge
- **Proof of Burn** – like Proof-of-Stake, except staked coins are burned

## VI. Proof-of-Stake

With proof-of-stake, **validators** are stakeholders with voting power proportional to the **economic stake** they have locked up. The assumption here is that someone with more stake is more incentivized to do things that will benefit the system and thus increase their economic stake.

**Chain-based PoS** chooses availability while **Byzantine Fault Tolerant PoS** chooses consistency.

### Weaknesses of PoS versus PoW:

- PoS is susceptible to corruption if over 33% of the network are malicious actors, whereas PoW requires over 50% malicious actors
- PoS tends to lead to a rich-become-richer problem where those who stake substantial portions of the total network will grow in proportion due to higher likelihood of being selected, and thus rewarded
  - If the larger players grow past 33% of network, poses a threat to validity

## VII. Federated Consensus

Federated consensus allows us to achieve explicit voting and censorship resistance, so that we can allow anyone to join but also protect the network against Sybil attacks.

If you don't trust certain nodes in the quorum, we can avoid having a central party choose the quorum for us by using a quorum slice, or subsets of a quorum that a particular node can trust. A **quorum slice** allows us to individually choose who we trust, and when multiple quorum slices overlap, we form **quorum intersections** and thus a larger quorum.

Federated consensus is powerful because of its decentralized control, low latency, and flexibility towards trust. Popular implementations of federated consensus include **Ripple** and **Stellar**.

Subscribe

### Readings

[Short Overview of Alternatives to PoW Opens in new window](#)

[Adventures in Galactic Consensus Opens in new window](#)

[Stellar Consensus Protocol Overview Opens in new window](#)

[CAP Theorem Overview Opens in new window](#)

[Raft Overview Opens in new window](#)

Video: [Software Powering Falcon 9 & Dragon, Simply Explained Opens in new window](#)

[Time, Clocks, and the Ordering of Events in a Distributed System Opens in new window](#)

[The Byzantine Generals Problem Opens in new window](#)

[The Part-Time Parliament Opens in new window](#)

[Paxos Made Simple Opens in new window](#)

[In Search of an Understandable Consensus Algorithm Opens in new window](#)

[Practical Byzantine Fault Tolerance Opens in new window](#)

[Proofs of Space Opens in new window](#)

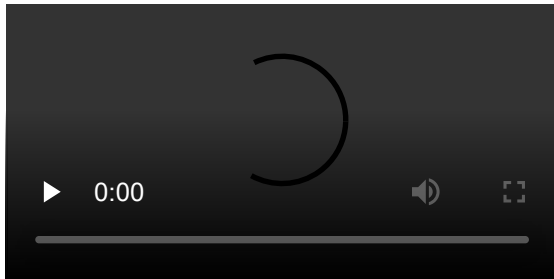
[Tendermint: Byzantine Fault Tolerance in the Age of Blockchains Opens in new window](#)

[Casper the Friendly Ghost Opens in new window](#)

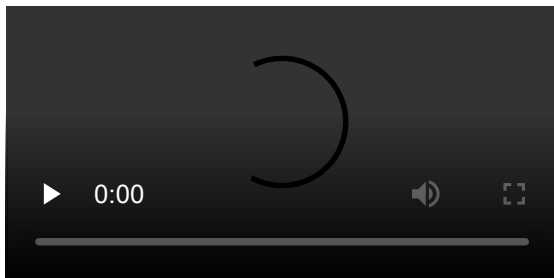
[Casper the Friendly Finality Gadget Opens](#)

## Securing Incentives: Cryptoeconomics & Proof-of-Stake

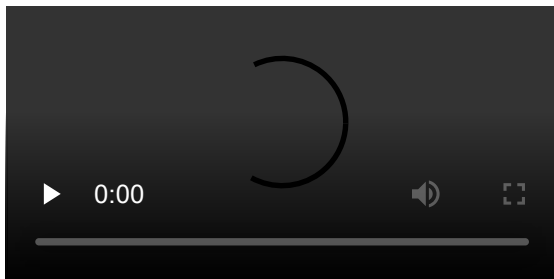
Welcome to Week 2



Intro: Cryptoeconomics

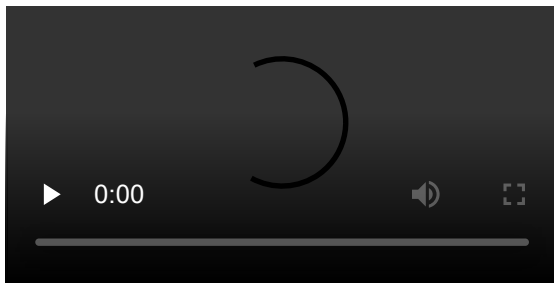


Cryptoeconomics

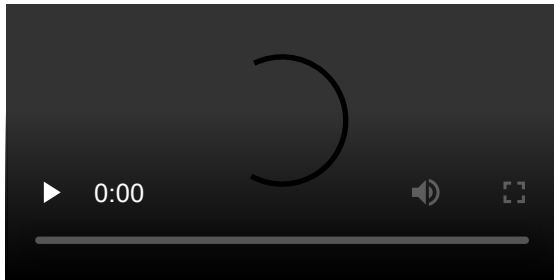


Intro: Cryptography

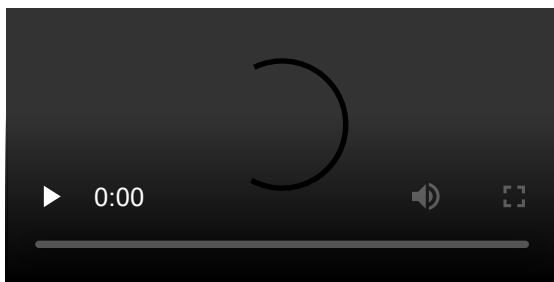
Subscribe



## Origins of Cryptography

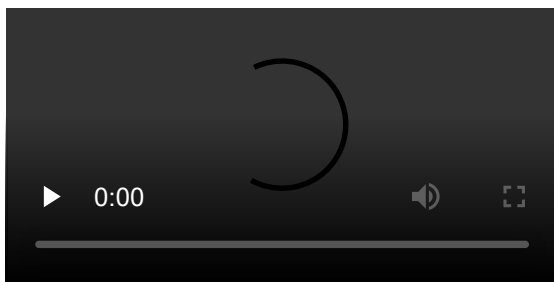


## Cryptographic Primitives

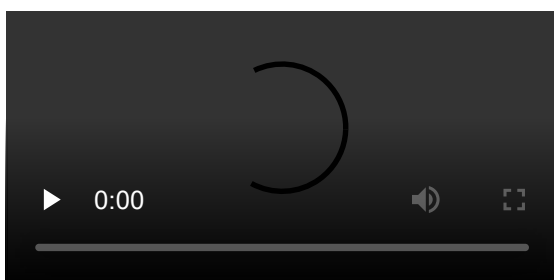


Subscribe

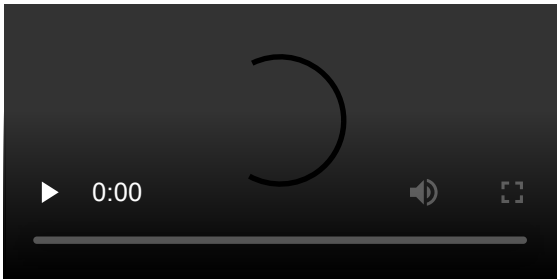
## Intro: Economics



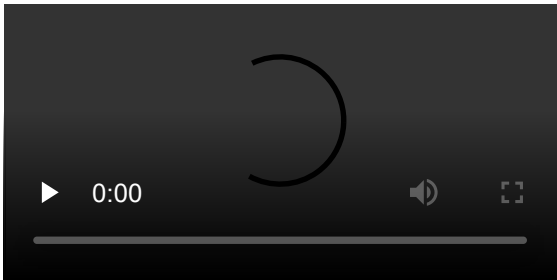
## Game Theory



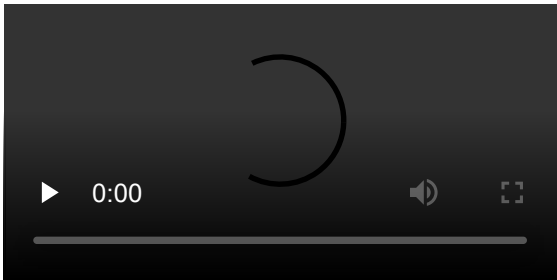
## Blacklisting



Intro: Proof-of-Stake

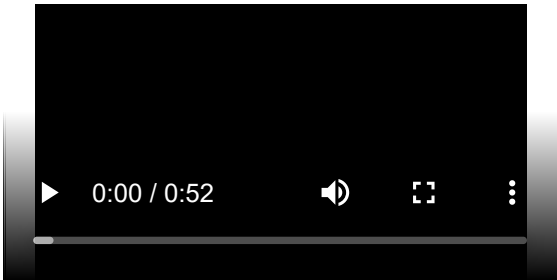


Proof-of-Stake

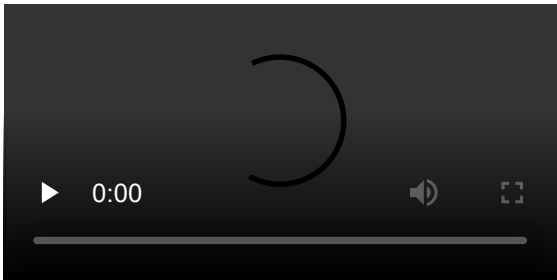


Subscribe

Intro: Attacks



Attacks



Text: Week 2 Summary

Written by Deven Navani and Nicholas Shen

## I. Cryptoeconomics

Economic principles help us to design a system so that actors are incentivized to make decisions in line with the goals of the greater good. We are able to **secure the future**. (e.g. block reward in Bitcoin and cost of mining to deter Sybil attacks)

Cryptography allows us to **secure the past** and ensure our decisions cannot be manipulated by observers (e.g. cryptographic signatures for authentication and hashes for immutability)

## II. Cryptography

**Cryptography** aims to secure the integrity and confidentiality of information.

The need for cryptography is especially important in distributed systems, where unknown actors are a potential threat to the secrecy and safekeeping of information.

**Encryption** is the process of transforming information into an unintelligible intermediary piece of information which can be transformed back into its original state with **decryption**. An early example of encryption was the Roman Empire's use of the **Caesar Cipher**, in which messages are encrypted by shifting letters to the right by a previously set amount.



Be aware of various cryptographic primitives (review from previous course):

Subscribe

- **Cryptographic hash functions**, used to capture the identity of information without revealing anything about the information itself
- **Digital signatures**, used to prove your identity and that you sent a particular message
- **Erasur codes** lower the 100% data availability requirement
- **Timelocks** allow for a message to be easily encrypted but take a longer amount of time to decrypt.

## III. Economics

Economics boils down to a fundamental question: how do you determine the best choice to make with your limited resources in order to maximize your profit? Economics also helps us to design a system so that everyone is incentivized to act in a certain way.

In **game theory**, we aim to deduce how an actor will act in a given situation. These decisions are influenced by the actions of others and the rewards and penalties associated with certain decisions. Therefore we aim to manipulate these factors.

In blockchain, **tokens** are used as economic incentives. Tokens are units of protocol defined cryptocurrency given out to miners and privileges miners can charge for. The assumption here is that the underlying objective for actors in a blockchain network is to maximize their profit, which equals their revenues minus their costs.

## IV. Proof of Stake

Proof-of-Stake is a particular type of consensus mechanism that assumes all voting power is tied to financial resources. Fundamentally, the idea is: the more tokens or currency an actor holds within a Proof-of-Stake system, the stronger the incentive for them to be good stewards of said system; if the system grows the wealthier the actor becomes. Thus in Proof-of-Stake, we give these individuals the most power as validators.

Major PoS implementations:

- **Tendermint** – First BFT-based PoS consensus mechanism, published in 2014
- **Casper the Friendly GHOST (CBC)** – a family of consensus algorithms designed from ground up i.e. Correct-by-construction, a proposed upgrade for the Ethereum network
- **Casper the Friendly Finality Gadget** – a Proof-of-Work and Proof-of-Stake hybrid; another upgrade proposed for the Ethereum network

## V. Attacks

Each proof of stake attack represents a scenario in which the incentives of an individual are not aligned with the incentives of a group, i.e. giving an unfair advantage to any single actor. Because the resource consumed is monetary value, bad actors need to receive an explicit monetary penalty with each attempted attack to keep the system in check.

If there was zero penalty, the expected profit of any given attack would be some number greater than zero, providing an incentive. By penalizing users for incorrect or malicious actions, the system hopes to bring the expected value to less than or equal to zero.

Subscribe

Examples of attacks:

- **Nothing-at-Stake**: voting in favor of every fork in hopes of maximizing one's rewards i.e. guaranteeing you will not miss the reward from the chosen branch; solution: slashing an actor if they are caught voting on multiple forks, or a less popular scheme penalizes incorrect votes; keep in mind that voting takes place using cryptographically identifiable/verifiable signatures.
- **Stake grinding**: attack where a validator performs some computation or takes some other step to try to bias the randomness in their own favor; solution: require validators to deposit their coins well in advance, and avoid information that can be easily manipulated as source data for randomness

**Weak subjectivity** is a problem for new nodes or nodes that have been offline for a long time; the node does not know which chain is the main chain; solution: introduce a "revert limit" – a rule that nodes must simply refuse to revert further back in time than the deposit length

Readings

[A Proof-of-Stake Design Philosophy Opens in new window](#)

Princeton Textbook Ch 8.5 Proof-of-Stake and Virtual Mining (link in Week 0 resources)

[A \(Short\) Guide to Blockchain Consensus Protocols Opens in new window](#)

[Consensus Mechanisms Explained: PoW vs. PoS Opens in new window](#)

[What is Cryptoeconomics Opens in new window](#)

[Long-Range Attacks: The Serious Problem With Adaptive Proof of Work Opens in new window](#)

[On Stake Opens in new window](#)

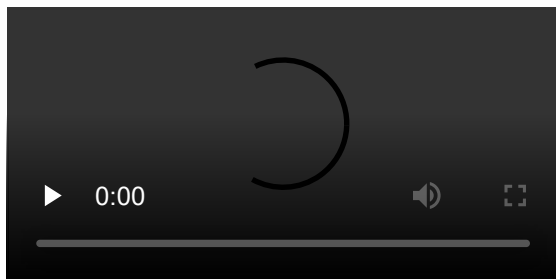
[Secret Sharing and Erasure Coding: A Guide for the Aspiring Dropbox Decentralizer Opens in new window](#)

[Introduction to Cryptoeconomics Opens in new window](#)

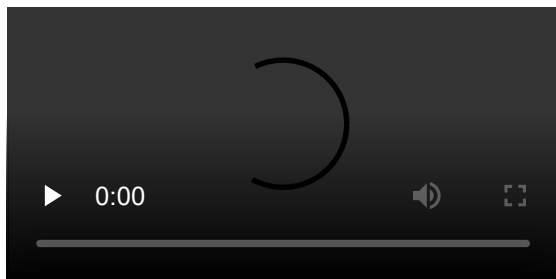
[EB105 – Vlad Zamfir: Bringing Ethereum Towards Proof-Of-Stake With Casper](#)

## Real-World Applications: Enterprise Blockchain

Welcome to Week 3



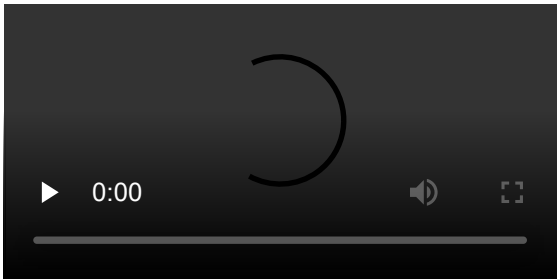
Intro: Enterprise Blockchain Overview



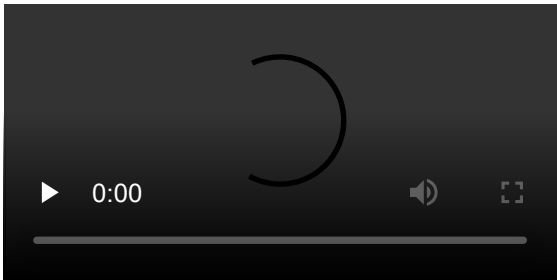
Enterprise Blockchain History

Subscribe

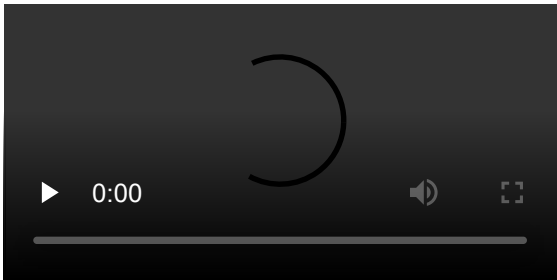




Enterprise Blockchain Overview

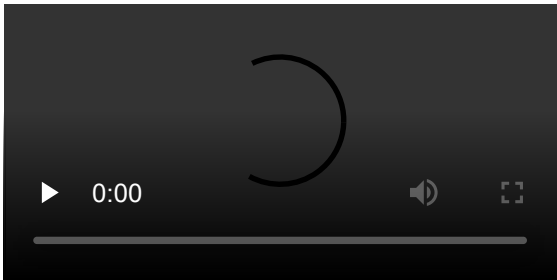


Intro: Enterprise Blockchain Platforms

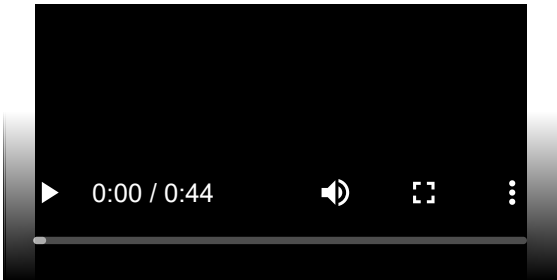


Subscribe

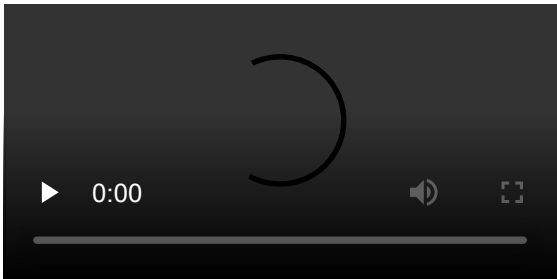
Enterprise Blockchain Platforms



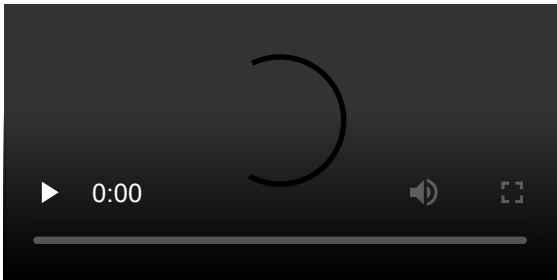
Intro: Use Cases & Industries



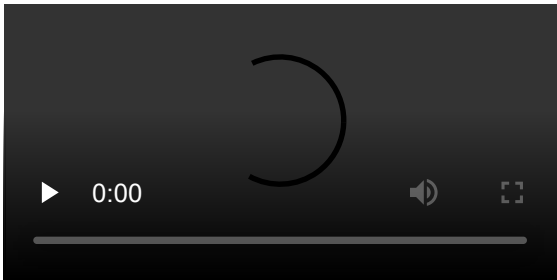
Use Case: Auto & Mobility



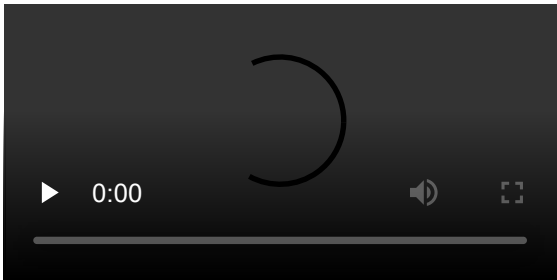
Use Case: Finance



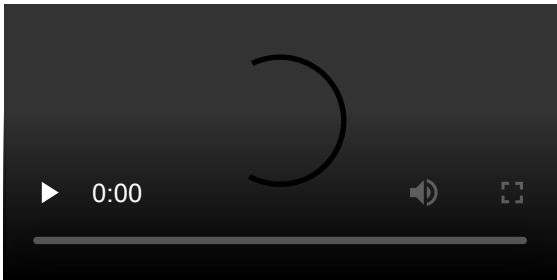
Use Case: Travel & Tourism



Use Case: Digital Identity

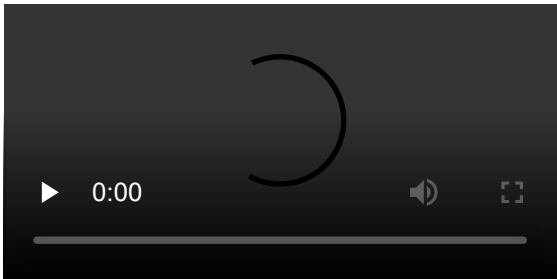


Use Case: Healthcare

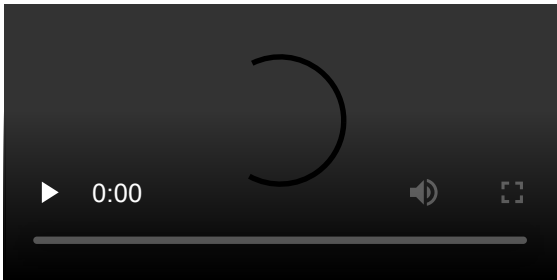


Use Case: Insurance

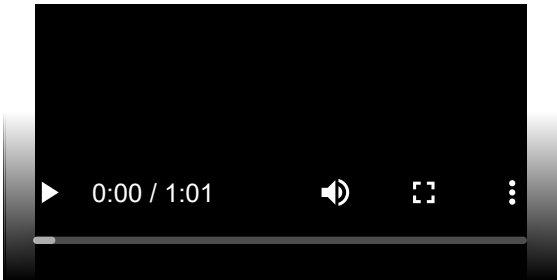
Subscribe



Use Case: Supply Chain

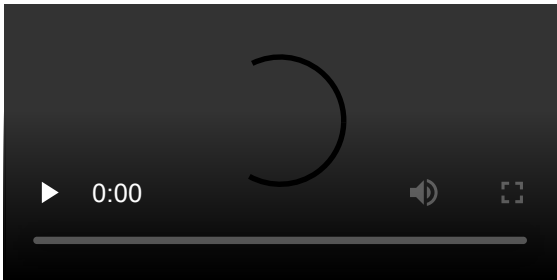


Use Case: IoT

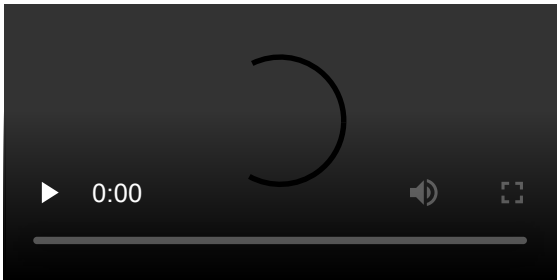


Subscribe

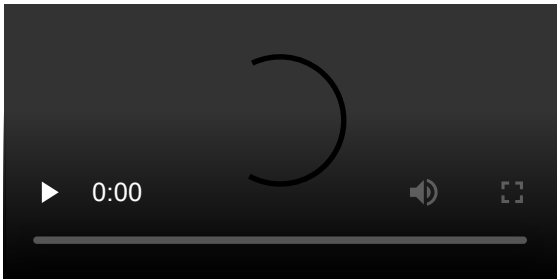
Use Case: Housing & Real Estate



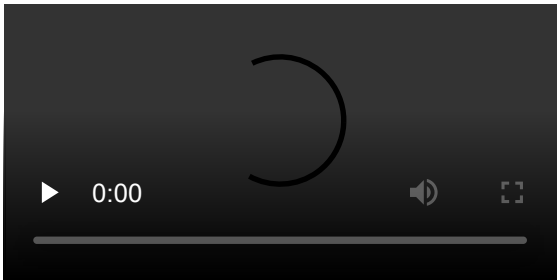
Use Case: Foreign Aid



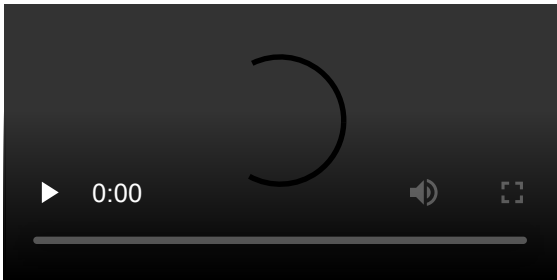
Use Case Generalizations



Intro: ICO Schemas & Culture

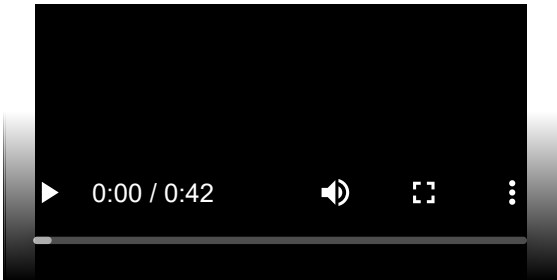


ICO Schemas & Culture

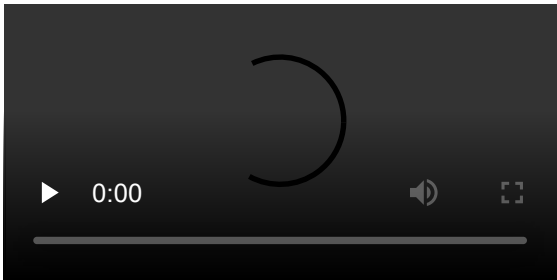


Subscribe

Intro: Regulation & Caveats



Regulation & Caveats




Text: Week 3 Summary

## I. Enterprise Blockchain Overview

As Bitcoin and blockchain technology matured, banks and corporations took interest in developing what are now known as **permissioned** blockchains and distributed ledgers. They aimed to “take the blockchain out of Bitcoin.”

Permissioned systems only allow trusted users into the system, allowing for a reduction of key properties pushed by public blockchains, resulting in systems with reduced levels of openness, no guarantee of trustlessness, and fewer incentives built into the protocol.

Primarily, enterprise blockchains of the time were used to solve issues in **coordination failures**, boost **horizontal integration**, and create **self-sovereign** decentralized networks.

 There are two main categories of databases: Centralized Databases and Distributed Databases. Within Distributed Databases, there is Distributed ledgers. Within Distributed ledgers, there is Blockchain

**Centralized databases** are run by a single entity (e.g. a company) that handles all requests and data processing.

 Central Database

**Distributed databases** are run by a group of storage nodes that are connected to each other and work to maintain a consistent overall view of the entire system. Nodes are able to fully trust each other in some systems (hence the solid lines connecting storage nodes.)

Subscribe

 Distributed Database

**Distributed ledgers** are a specific type of distributed database in which the information is organized chronologically, mimicking a traditional ledger. Most often, storage nodes may not fully trust each other (hence the dotted lines in the diagram below). Instead, they must implement some form of consensus protocol to have a consistent view of the system.

 Distributed Ledger

Distributed ledgers that specifically implement a chain of blocks in their protocol are known as **blockchains**.

Blockchains exist in three broad categories, depending on their access types: public, consortium, and private blockchains. Together, consortium and private blockchains are known as permissioned ledgers, since they require some level of permission granted – as opposed to openly readable and writable public blockchains.

## II. Enterprise Blockchain Platforms

There exist many enterprise blockchain platforms today – too many to mention in detail in this summary. The key things to look for when evaluating whether a particular enterprise

blockchain platforms is right for a particular use case are:

1. Enterprise blockchain platforms usually specialize in particular use cases, or have been used in the past to address certain use cases
2. As they specialize in particular use cases, they make usage assumptions that affect overall system scalability, security, and decentralization
3. These properties are affected by the underlying consensus mechanism(s) an enterprise blockchain platform supports

### III. Use Cases & Industries

Enterprise blockchains are being used today in a number of different use cases, including: auto/mobility, finance, travel/tourism, digital identity, and supply chain.

In general, the essential properties of a good blockchain use case are that:

1. Blockchain is not only viable, but is **necessary**. Otherwise, it's hard to justify a blockchain's low "efficiency"
2. Blockchain is used to **solve coordination failures**. Blockchain could be used to create arbitrary incentive structures and enable the cooperation of an untrusting consortium of companies and entities.
3. Blockchain aids in **horizontal integration**. Since data is now stored in a logically centralized blockchain, we can combine data silos and enforce a common API and data standard.
4. Blockchain achieves **pure decentralization**. This is not as relevant to enterprise blockchains, but blockchain in general (public ones) can be used to avoid centralized corruption.

Subscribe

Always keep in mind the advantages of centralized database solutions, and think of whether they, or a subset of blockchain technology, could be used to solve your business need – rather than an entire blockchain.

### IV. ICO Schemas & Culture

An **initial coin offering (ICO)** is a novel, "unregulated" means of raising funds for a blockchain startup.

ICOs are meant to allow developers to monetize open-source software despite the traditional incentive to make software proprietary. Additionally, it gives blockchain projects a much larger source of investors than only a relatively smaller set of VCs and other accredited investors.

However, ICOs also come with caveats. Because of a lack of regulation, scams are more capable of making their way into the view of investors, less doable when all investments were first screened either by VCs or government bodies, forcing investors to do more of their own due diligence. Additionally, many ICOs raise so much money that they have no incentive to actually finish up the project, leading to incentive misalignments.



Ryan Gosling, famed actor, also alleged graphic designer! Image used for ICO team of MIROSKII, fake cryptocurrency project

## V. Regulations & Caveats

As the world has never seen anything like blockchain before, there are still few regulations to specifically handle cryptocurrency and blockchain related matters.

First, because cryptocurrencies are inherently deregulated, they not only fail to abide by, but also may attempt to circumvent, laws such as anti money laundering (AML) laws and know your customer (KYC) regulations, leading to conflicts between regulatory bodies and cryptocurrency projects and exchanges. Exchanges are required to acquire licenses, such as a money transmitter license or a New York BitLicense in order to provide services. Some governments have taken steps towards regulating cryptocurrencies and blockchain, for better or for worse. Vermont and Arizona have declared that portions of the information on a blockchain can be considered legal evidence in court, but some countries have taken steps to restricting access to cryptocurrencies.

### Readings

[Blockchain in Enterprise: How Companies are using Blockchain Today Opens in new window](#)

[Enterprise Blockchain is Ready to Go Live Opens in new window](#)

Subscribe

[Enterprise Blockchain Ready for Breakout Opens in new window](#)

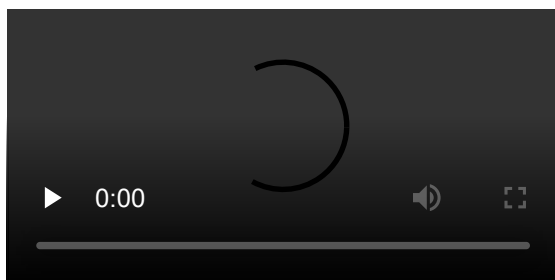
['Decentralized Bank' ICO Miroskii's Entire Team Is Phoney Opens in new window](#)

[Quorum: Ethereum for enterprise applications Opens in new window](#)

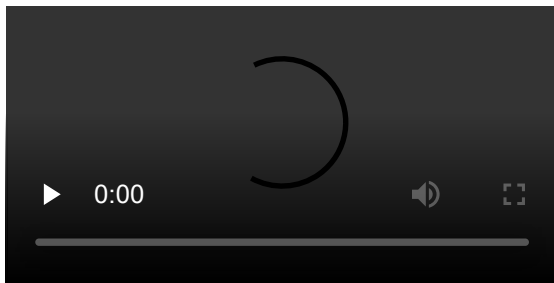
[All You Need To Know About Initial Coin Offerings](#)

## Cryptocurrency for the Masses: Scaling Blockchain

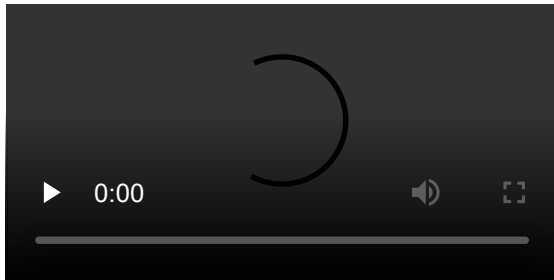
Welcome to Week 4



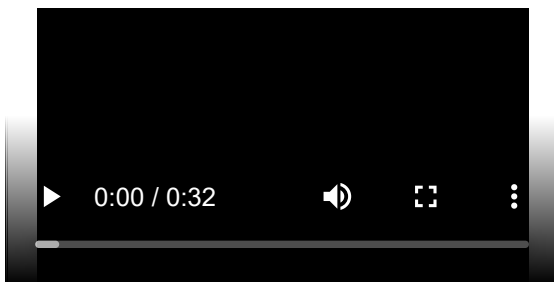
Intro: Scaling Background



### Scaling Background

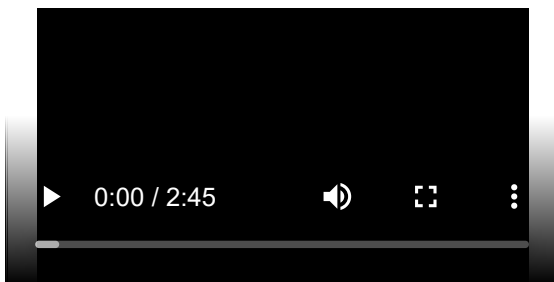


### Intro: Vertical Scaling On-Chain

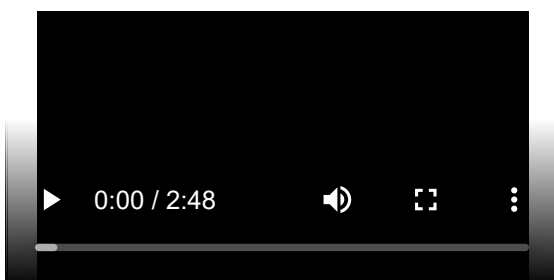


Subscribe

### Naive Solution

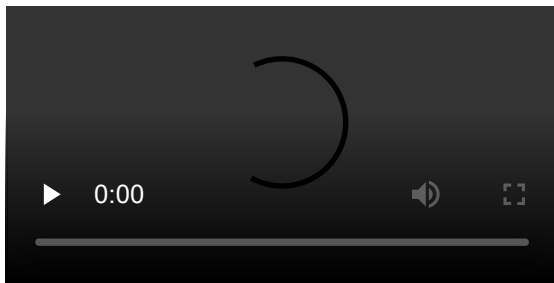


### Decrease Block Time

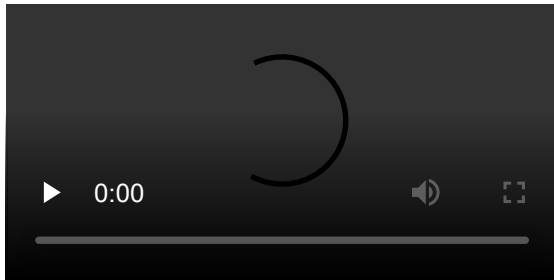


### Increase Block Size

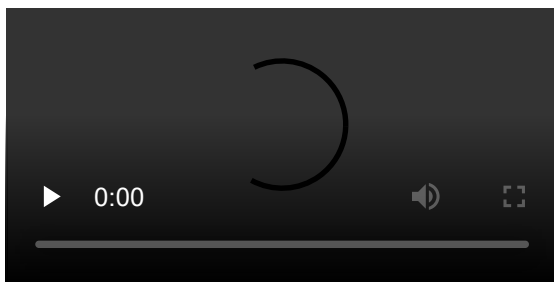




### Decrease Transaction Size

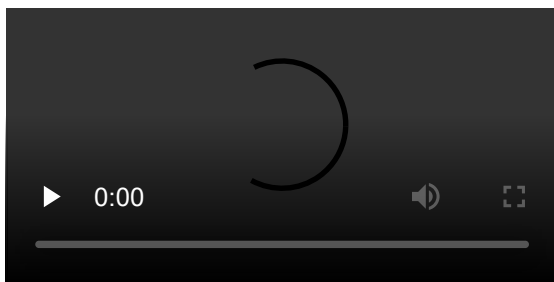


### Intro: Vertical Scaling Off-Chain

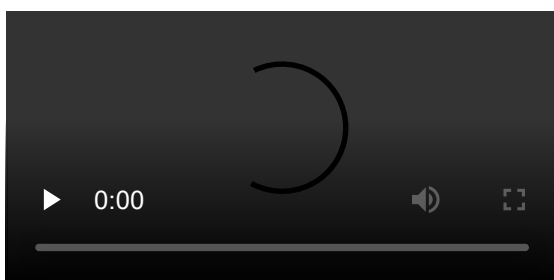


Subscribe

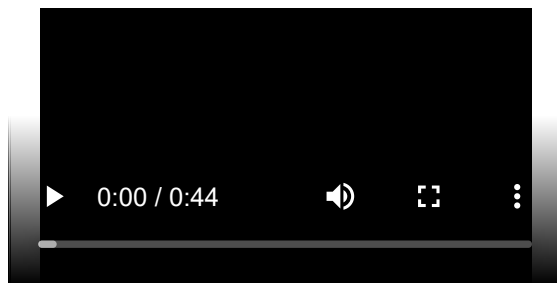
### Payment Channels



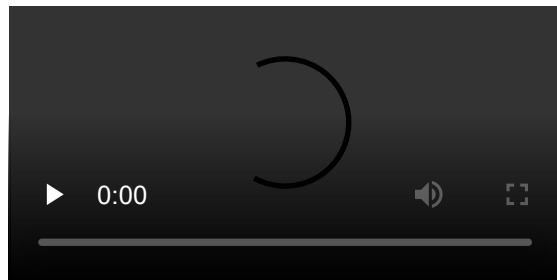
### Lightning & Raiden



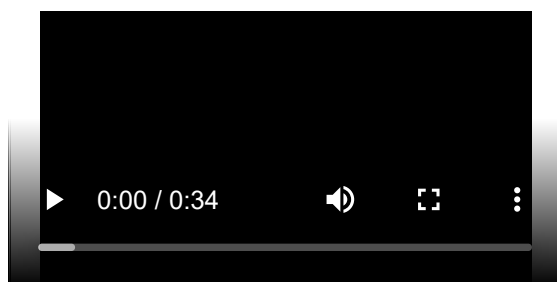
### Intro: Horizontal Scaling



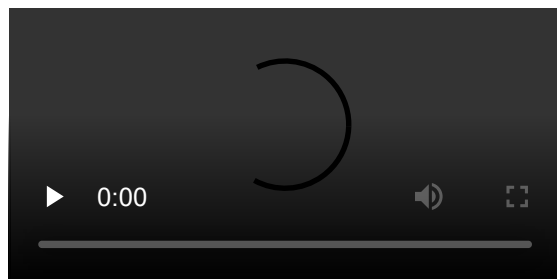
## Sharding & Sidechains



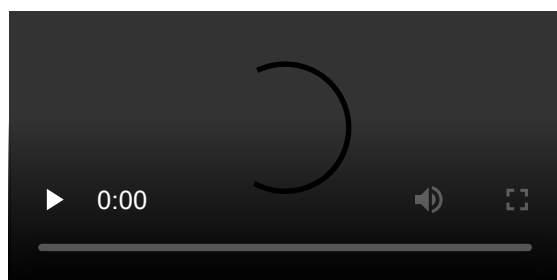
## Intro: Advanced Scaling & Generalizations



## Advanced Scaling



## Generalizations



## Text: Week 4 Summary

Subscribe

## I. Background

Modern day public blockchains have been victims of their own success. Bitcoin and Ethereum especially are having scalability issues in that they aim to be global networks able to support global-scale transaction volumes, but currently both perform subpar in the transaction throughput.

Fundamentally, scaling solutions can either increase the transaction volume, or decrease the block time. This is self evident as scalability is measured in a blockchain's achievable TPS (transactions per second.)

Going forward, we can classify blockchain scaling solutions two ways. The first is a rough comparison with traditional cloud architecture scaling classifications: horizontal, vertical, and diagonal. Secondly, there are the blockchain-specific scaling classifications: layer 1 (on-chain) and layer 2 (off-chain).

## II. Vertical Scaling

### TPS calculation

Bitcoin processes less than 10 transactions per second, and without any scalability upgrades, it's bound to stay at low TPS. Looking at how we calculate TPS in the first place, namely in the rough dimensional analysis above, we can see that the fields we can attempt to modify in efforts to create new scaling solutions are:

Subscribe

1. Block time
2. Block size
3. Transaction size

These parameters are all built into a blockchain system itself, and tuning these parameters directly constitute as layer 1 scaling solutions.

### II.I. Decrease Block Time

We can't simply decrease the block time of a blockchain system, since that would result in a higher rate of naturally occurring forks, reducing system security. This is because while block time decreases, the time to propagate a block remains the same.

### Time to broadcast block fixed while block creation time decreases

Ethereum has dealt with this problem historically by employing the GHOST (Greedy Heaviest Observed SubTree) protocol. With the GHOST protocol, miners no agree on the longest chain to be canon (as in Bitcoin), but rather the chain with the most "weight", where weight is a value calculated by both a chain's length and the number of uncle blocks it has.

### II.II. Increase Block Size

Increasing block size would improve a blockchain's TPS. Since a block can now contain more transactions, it would also lower transaction fees.

However, as with decreasing block time, there are some side effects. For one, increasing block size would imply hard forking, and depending on the community, this could be a less than pleasant experience. It would also make the blockchain grow in size at a much faster rate – a problem decreasing the block time also faced. And finally, increasing the block size is most likely not a one-time fix, since the scalability boost is only linear. The block size might need to be increased in the future again, leading to a “slippery slope” type of debate.

### II.III. Decrease Transaction Size

Segregated Witness (SegWit) was an upgrade to Bitcoin that move transaction signatures from within the transaction to a separate structure at the end of the transaction, called the segregated witness. To non-SegWit nodes, this would be a decrease the effective transaction size since they wouldn't know to read into the segregated witness.

Non-SegWit nodes would see a transaction without a signature, but would mark the transaction as valid. SegWit nodes on the other hand would know to read into the segregated witness, and would verify it using the signature.

SegWit was originally designed to solve transaction malleability in Bitcoin. It also is implemented with a soft fork, and results in a smaller blockchain size. However, SegWit is only one time linear scalability boost.

Subscribe

Recursive SNARKs also decrease transaction size. Instead of storing transactions themselves in the blockchain, we could instead store proofs that these transactions have indeed occurred, and the final balance sheet of who owns however much cryptocurrency. This leads to efficiency gains by decreasing transaction size, and also because machines can verify proofs within milliseconds. However, currently, a trusted environment setup is required in order to produce these style of proofs. And proof generation in practice is very costly.

### III. Vertical Scaling Off-Chain

Given that the speed of a blockchain limits its scalability, we can consider entirely removing the more costly operations off the chain and only publishing when we require a global sense of truth.

Payment channels in Bitcoin could be implemented using HTLCs (hash time lock contracts), and could move transactions off the main Bitcoin blockchain and onto side chains. If Alice and Bob transact often, perhaps it makes sense for Alice and Bob to construct a private payment channel, where they conduct their transactions off-chain. Only when they want to settle their final balances do they post back to the main blockchain. This allows Alice and Bob to still conduct their transactions as they do, but the main blockchain only has to store Alice and Bob's initial and end balances.

The idea behind the Bitcoin Lightning Network is to create a network of payment channels

 Payment Channel Network. Alice to Bob to Eve to Charlie

In the diagram above, Alice can pay Charlie without having a payment channel to Charlie directly, so long as there is a path from Alice to Charlie through the payment channel network.

Ethereum has a similar scalability solution in the works, appropriately named Raiden.

Payment channels and payment channel networks would allow us to keep many transactions off chain, delegating payments to simple bookkeeping. Since the main blockchain only sees the start and end balances of the parties in a payment channel, we can keep a majority of transactions off chain: scaling Bitcoin from under ten transactions to potentially hundreds of thousands of transactions.

Some problems include having to lock up capital in order to initiate a payment channel, and centralization concerns of payment channel networks converging to hub-and-spoke topologies.

## IV. Horizontal Scaling

Sharding is database scaling strategy that breaks up a monolithic database into “shards”, each a separate database that contains data from a subset of the original database, whose union is the original database. The same idea can be applied to blockchain, and is currently one of the active areas of research in Ethereum research.

Subscribe

The idea translated to blockchain implies that not every node keeps track of every block. It would be a layer 1 horizontal scaling solution. We could have multiple blockchains running in parallel, each containing a subset of all transactions. Issues currently being researched include the classification of various nodes in a sharded blockchain system (e.g. nodes that see a single shard vs nodes that see all shards), cross-shard communication, and defenses against single shard takeovers.

Sidechains are the idea that you can create multiple side chains for different purposes that plug into a main chain, effectively decreasing the traffic on it.

This does separate hashing power across multiple chains, which raises security concerns.

Here is an example of a sidechain setup:

 Side chains

Source: <https://blockstream.com/technology/> Opens in new window

## V. Advanced Scaling & Generalizations

Ethereum's Plasma can be seen as a diagonal scaling solution, since it enables horizontal scaling by implementing side chains and vertical scaling by increasing their speed through Tendermint and alternative consensus mechanisms. The security of off-chain transactions is derived from the root chain, the main source of truth within the system.

FourthState, a team comprised of Blockchain at Berkeley's members, wrote an implementation of Plasma using the Cosmos SDK, enabling further flexibility and scalability.

Blockchains have 3 main abstraction layers, from top to bottom:

- The application layer processes transactions and updates the state of the system
- The consensus layer makes sure the entire network agrees on transactions and updates to the database
- The networking layer makes sure all nodes get updates within a reasonable amount of time

The purpose of the Tendermint project is to provide the networking and consensus layers so that arbitrary applications could be built on top of it. Tendermint is the consensus "engine" of the Cosmos network, which aims to make blockchains interoperable and scalable.

The following table summarizes the scaling solutions we have learned, categorized by 2 different methods. Layer 1 and Layer 2 specify whether solutions are built on-chain or off-chain. Solutions can also scale vertically or horizontally.



Vertical and Horizontal, Layer 1 and Layer 2 scaling solutions

Subscribe

## Readings

[Blockchains don't scale. Not today at least. But there's hope. Opens in new window](#)

[What is the Lightning Network? Opens in new window](#)

[How to Scale Ethereum: Sharding Explained Opens in new window](#)

[What is SegWit? Opens in new window](#)

[Tendermint and Cosmos Opens in new window](#)

[The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments Opens in new window](#)

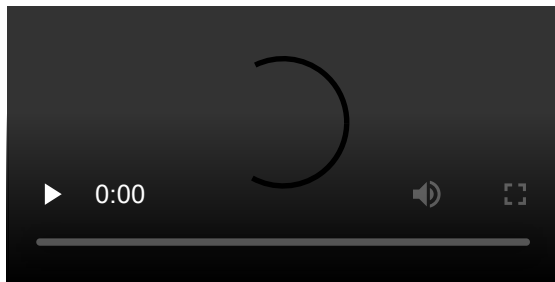
[Plasma: Scalable Autonomous Smart Contracts Opens in new window](#)

[Minimal Viable Plasma Opens in new window](#)

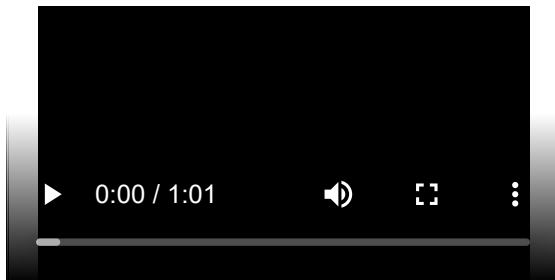
[Cosmos: A Network of Distributed Ledgers](#)

# The Fight for Privacy: Anonymity, Mixing & Altcoins

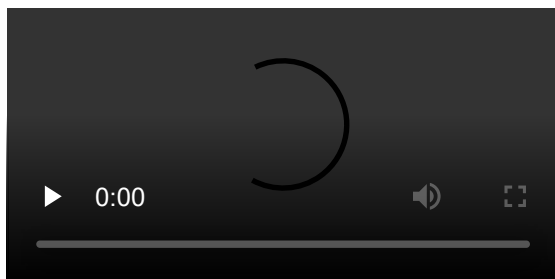
Welcome to Week 5



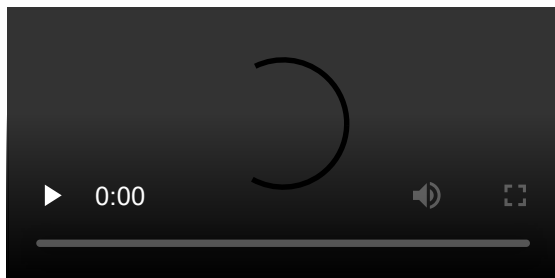
Intro: Anonymity Basics



Anonymity Basics

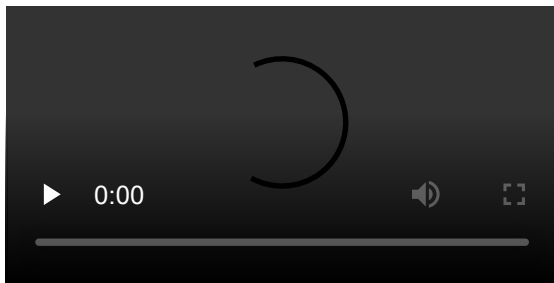


Intro: Deanonymization

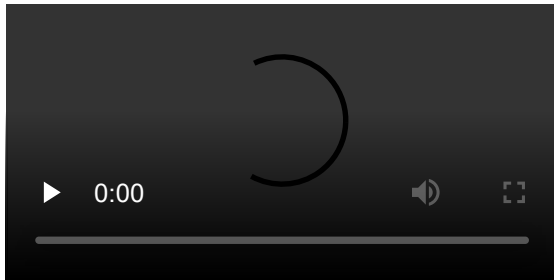


Deanonymization

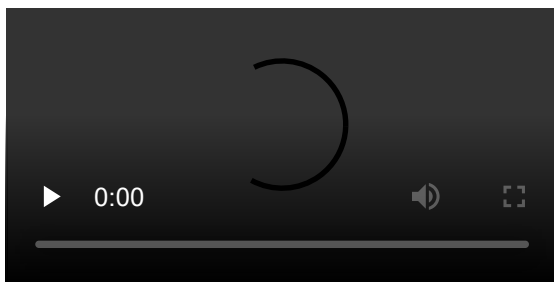
Subscribe



### Intro: Anonymity Through Mixing

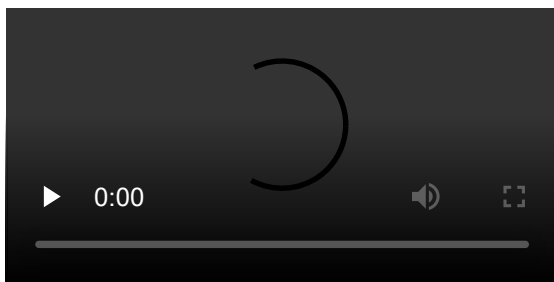


### Mixing Basics

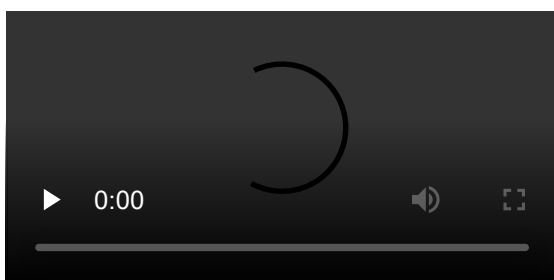


Subscribe

### Decentralized Mixing

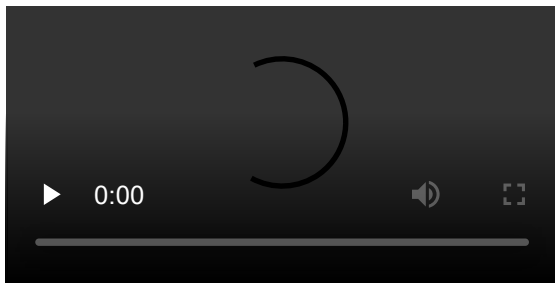


### Fair Exchange Mixers

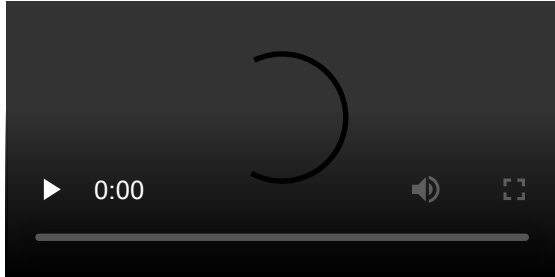


### Intro: Anonymity Through Altcoins

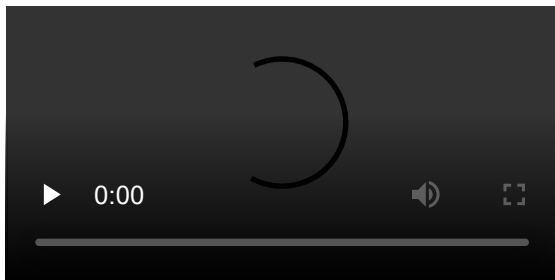




### Privacy Focused Altcoins

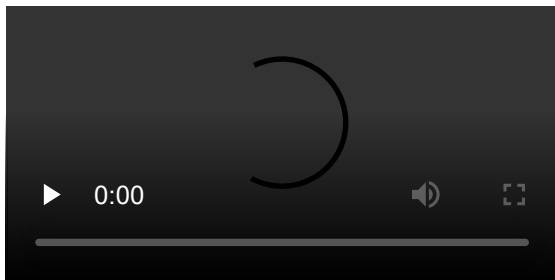


### Intro: Advanced Anonymity & Generalizations



Subscribe

### Advanced Anonymity & Generalizations



### Text: Week 5 Summary

#### I. Anonymity Basics

It's a common misconception that anonymity in cryptocurrencies is only useful for buying drugs or participating in illegal activities. The truth is, everyone benefits from anonymity. Anonymity helps in providing crucial properties of currency. For example, **fungibility** is the idea that every unit of a currency must be equal in value to every other unit. Cash is fungible because it is anonymous – a dollar is a dollar. It may also be dangerous if information about a user's identity/assets are made public.

Most blockchains are pseudonymous by default, not anonymous. With their sights set on decentralization first and foremost, the original designers of Bitcoin chose to build a distributed database in which everyone (all full nodes) stores the Bitcoin blockchain. While this grants us decentralization properties, the fact that records of all transactions are now public is concerning, especially for those who do not actively adhere to best practices, such as generating a new pseudonym for each transaction. Additionally, the privacy of users can be severely compromised if their virtual identities, or their **pseudonyms**, are somehow linked to their real ones — this is known as **linking**.

And sometimes, as hard as one might try to make a particular technology secure, private, or anonymous, human factors make all of this very difficult.

## II. Deanonymization

The goal of deanonymization is to link your online identity (pseudonym) to your real identity through analysis and heuristics.

One technique is **Transaction Graph Analysis**, the process of constructing a **transaction graph** (pictured below) where each node is a pseudonym and each edge is a transaction conducted between pseudonyms.

This graph can be analyzed by **clustering**, or attributing a group of pseudonyms to the same real world entity. The first part of clustering, actually creating the clusters, can be performed using various heuristics, two of which are **merging transaction outputs** and **change addresses**. In regards to the second part, linking clusters with their real world identities, various tactics include:

Subscribe

- Tagging by transacting (useful for tracking businesses)
- Relying on carelessness (useful for tracking individuals)
- Service providers, who use data analytics to discover real identities

**Taint analysis** can be used to tag “bad” addresses and trace their associated activity throughout the network (**taint** is the percentages of funds received by an address that can be traced back to another address).

## III. Anonymity Through Mixing

An **anonymity set** is the set of pseudonyms between which an entity cannot be distinguished from their counterparts. The goal of **mixing** is to maximize the size of this anonymity set, because a larger anonymity set means it is more difficult to associate pseudonyms with real world identities.

### Centralized mixing solutions:

- **Third Party Protocol (TPP)**: Of course, as with any centralized service, TPP faces the issue of being a single point of failure. Additionally, if the only UTXOs being sent to the

centralized mixer are dirty coins, the outputs for later users will only be dirty coins unless there are enough clean coins being cycled through the slush fund.

- **Altcoin exchange mixing**, which works by sending dirty funds through several layers of altcoin  $\Leftrightarrow$  altcoin exchanges to obfuscate the money trail. This is less centralized than TPP. In this case, there is no mixing fee, but rather the sum of the exchange fees between each cryptocurrency used. Advantages of this technique include better plausible deniability and increased tracing difficulty, while disadvantages include the chance that exchanges may reveal the links between your inputs and outputs.

### Decentralized mixing solutions:

- **CoinJoin** was one of the earliest decentralized mixing protocols proposed, all the way back in 2011. It used n-of-n multi-signature transactions to mix coins together between n peers. The problems were that it assumed there was some central “mix facilitator” with a central server coordinating all the users, in addition to lacking plausible deniability and DoS resistance.
- **CoinShuffle** improved upon CoinJoin by introducing **decentralized mixnets**, which allow a set of users to pool together inputs from a group without revealing which input was submitted by which user. Additionally, it used an “Accountable Anonymous Group Messaging” protocol known as Dissent to coordinate users. Though this prevents attacks such as traffic analysis and deanonymization by a mix facilitator, it is still susceptible to all the drawbacks associated with CoinJoin.
- **JoinMarket** sought to fill a liquidity market of mixable coins for a fee, allowing anyone to contact this service to mix their coins. However, this approach does not provide a large anonymity set and was claimed to be deanonymizable with \$32,000 USD alone.
- **CoinParty** aimed to provide efficient mixing with plausible deniability and a larger anonymity set via escrow addresses and threshold signatures. However, this comes at the cost of some protocol security, as any  $\frac{2}{3}$  of the users colluding can steal funds from all the other users.

### Fair exchange mixers:

These are built upon the traditional **fair-exchange** protocol so that no trusted third party is needed.

- **CoinSwap** – uses hash-locked, 2-of-2 multi-signature transactions to securely swap coins without linking transactions. While this process is trustless and provides better plausible deniability, it comes at the cost of inefficiency and is also insecure against mix-passive intermediary.
- **XIM** – Uses untrusted intermediary to create fair-exchange mixer. It prevents Sybil and DoS attacks by enforcing fees to use service, but it requires several hours to run because of group-forming protocol.
- **BSC** – Builds upon XIM to allow users to skip group-forming protocol with anonymous fee vouchers. However, it is not supported on the Bitcoin protocol due to insufficient scripting capabilities.

- **TumbleBit** – implements an “RSA evaluation as a service” protocol to make BSC possible on the Bitcoin blockchain.

#### IV. Anonymity Through Altcoins

There exist altcoins with protocols where privacy is built in so that there is no suspicion towards any individual user. Examples include:

- **DASH** – employs a network of Masternodes to perform privileged actions (voting on proposals, confirming transactions instantly, and mixing the coins of all network participants)
- **Monero** – provides guarantees on transaction untraceability and unlinkability
  - *Untraceability*: for any incoming transaction, all possible senders are equiprobable
  - *Unlinkability*: for any two outgoing transactions, it is impossible to prove that they went to the same person
- **Zcash** – transactions reveal nothing about input and output addresses, using **zero-knowledge Succinct Non-interactive ARguments of Knowledge** (zk-SNARKs)

#### V. Advanced Anonymity & Generalizations

**Mimblewimble** proposes a more scalable and private cryptographic protocol than that of Bitcoin.

In regards to privacy, MimbleWimble supports **Confidential Transactions** (proposed by George Maxell), where all values in a transaction are encrypted with “blinding factors,” or secondary elliptic curves for increased privacy. MimbleWimble also bundles multiple transactions into larger transactions – this is done to scramble inputs and outputs.

Subscribe

MimbleWimble is currently under active development and its most popular implementation is called Grin.

MimbleWimble sacrifices some of Bitcoin’s advanced functionality (namely Bitcoin script) for anonymity and scalability. As with all other blockchain platforms, this brings us back to our trilemma – to enable the increase in privacy, there have to be sacrifices with regards to scalability and decentralization.

#### Readings

[Is Bitcoin Anonymous? A Complete Beginner’s Guide Opens in new window](#)

[Bitcoin Transactions Aren’t as Anonymous as Everyone Hoped Opens in new window](#)

[CoinJoin: Combining Bitcoin Transactions to Obfuscate Trails and Increase Privacy Opens in new window](#)

[Lord Voldemort Is Trying to Save Bitcoin Opens in new window](#)

[TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub Opens in new window](#)

[An Empirical Analysis of Traceability in the Monero Blockchain Opens in new window](#)

[How to Reveal a Secre Opens in new window](#)

[Mimblewimble Opens in new window](#)

[Confidential Transactions Opens in new window](#)

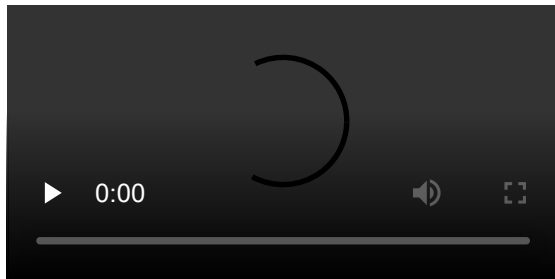
[Bulletproofs: Short Proofs for Confidential Transactions and More Opens in new window](#)

[MimbleWimble Explained Opens in new window](#)

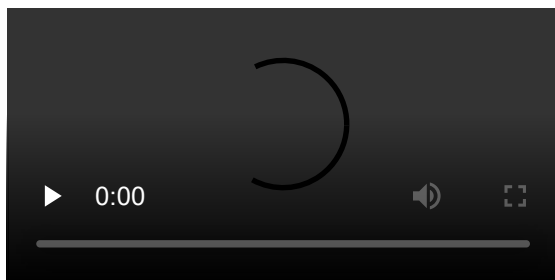
[Introduction to MimbleWimble and Grin](#)

## A Blockchain Powered Future

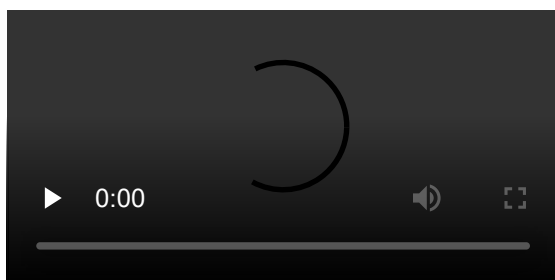
Welcome to Week 6



Intro: Program Summary

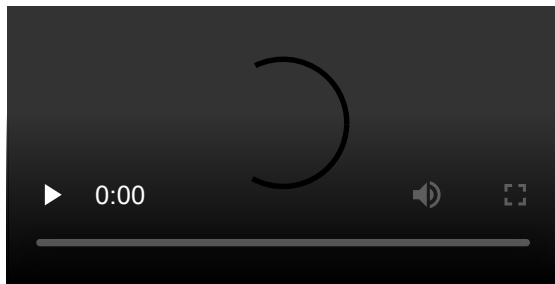


CS198.1x Bitcoin and Cryptocurrencies

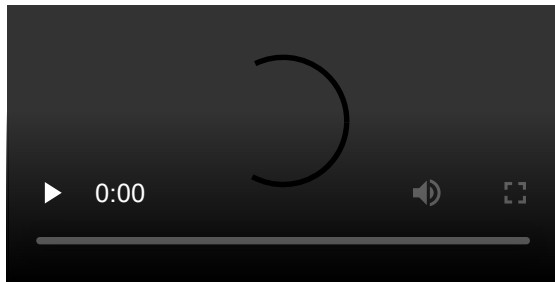


Subscribe

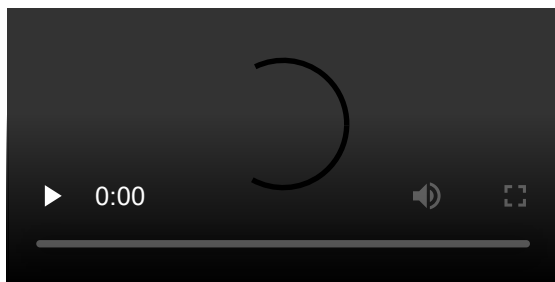
## CS198.2x Blockchain Technology



### Intro: People's Pick & FAQ



### People's Pick: Cryptocurrency Mining



Subscribe

### Frequently Asked Questions

Many frequently asked questions in our discussion board have been about the mechanics and theory of Bitcoin and cryptocurrency technologies. Such questions have been answered extensively in CS198.1x. Course staff have been making FAQ videos; you can find a [YouTube playlist here Opens in new window](#).

Here are some of the more recent FAQ for this course:

**Q:** What is in a block?

**A:** Many students who have not taken CS198.1x have been curious about the actual contents of a block, especially in Bitcoin, following our discussion of SegWit and other scalability solutions. In Bitcoin, a block contains: a block header, block size, transactions, transaction counters, and a magic number. We studied the block header extensively in lecture 3 of CS198.1x. It contains a version number, previous block header hash, merkle root, timestamp, target difficulty, and a nonce. These values are important for ensuring tamper-evidence. As

the goal of blockchain is to ensure certain innovative properties of blockchain, many other blockchains have similar structures.

**Q:** What is “hashing”?

**A:** In our courses, we’ve explained “hashing” to be the process of solving proof-of-work partial preimage hash puzzles. It’s best to think of hashing then as a lottery, in which the more you spend, the more likely you are to win. Or perhaps like throwing darts blindfolded – you can’t really aim your throws, so to increase your chances of getting a near-bullseye, simply increase your throwing rate. Related questions have been about effective hash rate, wasteful hashing, and hash functions in general. Course material can be found in CS198.1x weeks 5, 4, and 3 respectively. It may help to get a general overview of why we need to expend computational resources in the first place. This is the topic of proof-of-work consensus and of the more general class of Nakamoto consensus (expanding on the lottery analogy), which we define in CS198.2x week 1.

**Q:** Why so much focus on Bitcoin?

**A:** You may have realized this in our program summary section, but the overarching narrative of both of our courses can be seen as modeling the gradual maturing and general sentiment about cryptocurrency and later blockchain technologies. This approach explains the necessary context in the blockchain space, which influences design choices. Our narrative starts with CS198.1x Bitcoin and Cryptocurrencies, explaining cryptocurrencies as the first use case for blockchain, and Bitcoin as the original inspiration.

Subscribe

Our aim was to explore both the technological and social aspects of Bitcoin, and towards the end of the first course reveal that intuition from Bitcoin’s design carries over to understanding other blockchains in general. While the name of the first course is “Bitcoin and Cryptocurrencies,” one of the primary motivations was to decouple the ideas of Bitcoin and cryptocurrencies from that of blockchain. And with something to refer back to such as a solid understanding of Bitcoin, it makes analysis of blockchain systems in general much easier.

**Q:** Can blockchain be used in \_\_\_\_\_ industry?

**A:** There are many factors to take into account when answering this question. In this course, we have aimed to teach a framework with which students can gauge for themselves whether or not blockchain is useful for a given use case or industry. It mainly reduces to the question: what fundamental properties and innovations of blockchain specifically does the use case leverage? These are all design considerations, and the entirety of our second course should be of use to you. Course material related to blockchain use cases can be found in CS198.1x week 6 and CS198.2x week 3.

**Q:** Is \_\_\_\_\_ blockchain platform any good?

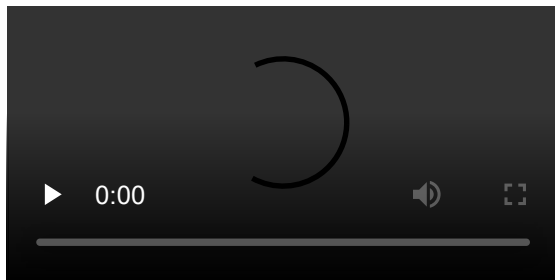
**A:** Again, our goal is to stay nonpartisan here. There will be invariably pros and cons and tradeoffs in the blockchain platform in question. These are all design considerations, and

hopefully our course material – especially CS198.2x week 3 – has empowered you to judge these parameters by yourself.

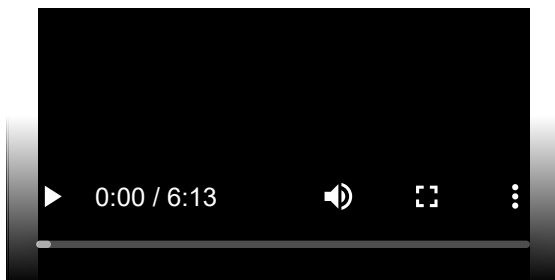
**Q:** Other. (How do I get my questions answered?)

**A:** We are actively supporting this course and engaging with students in the discussion boards. There are a lot of students in this course though, so the best way to get a question or concern addressed is to first scan the discussion boards to see if someone has asked something similar, upvoting the post if there is one, posting if there is not, commenting, etc. As a rough analogy to what we've learned throughout this course and last, the posts with the most "work" (upvotes, comments, activity) will be more likely to be addressed first.

Intro: Thought Experiment

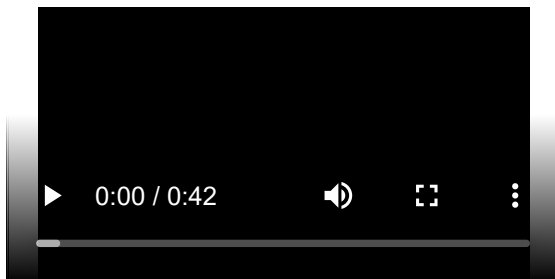


Blockchain Thought Experiment



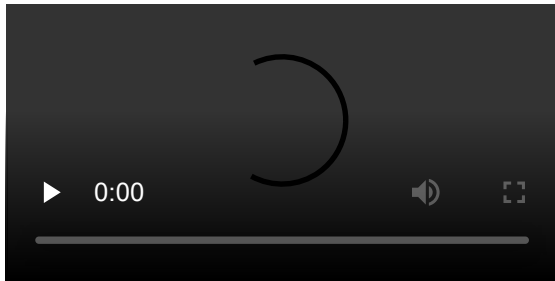
Subscribe

Intro: Getting Involved



Getting Involved





## Resources

The most accessible resources for getting more involved in the blockchain space can be found right here in edX. The goal of both of our courses in the Blockchain Fundamentals program was to provide students with a baseline understanding of blockchain – its design and context – surmounting the blockchain space’s traditionally high learning curve. The discussion boards have been a great place for students to not only get their questions and concerns resolved, but also to meet and network with other students. For example, some students have set up mailing lists, chat rooms, and even organized meetups – all from the discussion boards. Such student-created resources are not official resources, but we encourage decentralization!

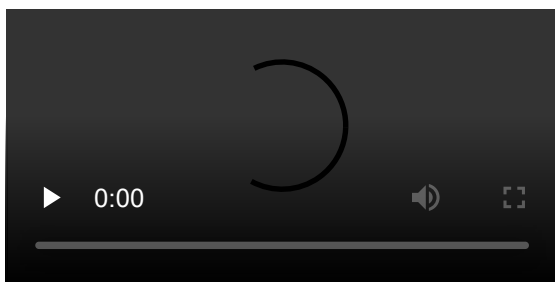
We have ported all of our educational materials over to our [Blockchain Fundamentals edX YouTube channel Opens in new window](#), for your quick reference in the future. Feel free to share video links with your friends and peers.

Now that you have finished this course, here are a list of other Blockchain at Berkeley resources. Some of you may have discovered these during the course run:

Subscribe

- [Blockchain at Berkeley website Opens in new window](#)
- Mailing list, located at the bottom of the website linked above
- [Discord chat Opens in new window](#)
- [Blockchain for Developers course \(Spring 2018\) Opens in new window](#)
- [Blog Opens in new window](#)
- [Blockchain at Berkeley YouTube channel Opens in new window](#)
- [Berkeley Bitcoin Meetup Opens in new window](#)
- [Facebook page Opens in new window](#)
- [Twitter](#)

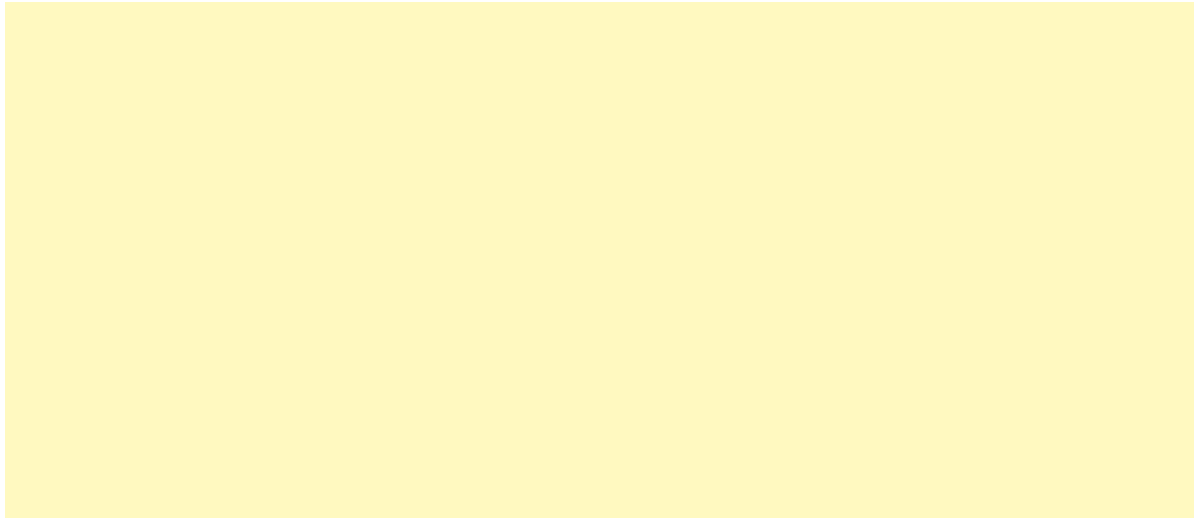
Thank You



[Read More](#)

[Join @LearnThingsOnline on Telegram](#)

Ads



## Leave a Reply

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

[Subscribe](#)

COMMENT

NAME \*

EMAIL \*

☐ Save my name, email, and website in this browser for the next time I comment.[Post Comment](#)

## LEARNTHINGS.ONLINE TELEGRAM GROUP

[Don't have Telegram yet? Try it now!](#)



Learn Things Online

65 members, 3 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.

Subscribe

## Stake your ADA with Wc

### Support the Cardano blockchain

We focus on reliability, security and  
responsiveness. Visit our website to l  
[wolverinestakepool.com](http://wolverinestakepool.com)

OPEN

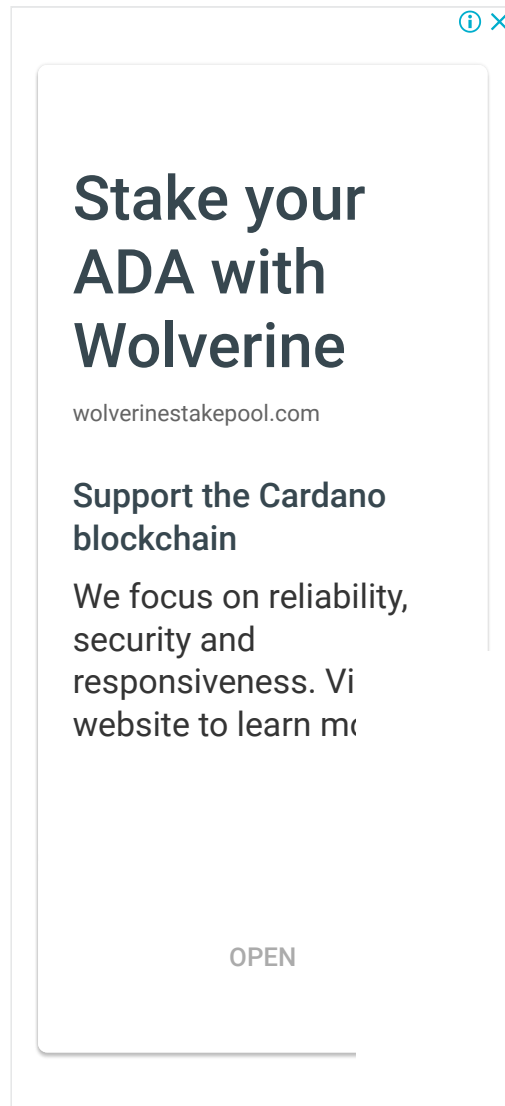


**RSS**

### 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.



The advertisement is a rectangular box with a light gray border. In the top right corner, there are two small icons: a blue circle with an 'i' and a blue 'X'. The main text is centered and reads: 'Stake your ADA with Wolverine' in a large, bold, dark gray font. Below this, the URL 'wolverinestakepool.com' is written in a smaller, dark gray font. Further down, the text 'Support the Cardano blockchain' is displayed in a bold, dark gray font. Below that, a paragraph reads: 'We focus on reliability, security and responsiveness. Visit our website to learn more.' At the bottom center, there is a button labeled 'OPEN' in a bold, dark gray font.

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



One place  
to create,  
collaborate,  
and connect



Try it free

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

©2020 LearnThings.Online