Build a Decentralized Network

FinTech

Lesson 18.3

Class Objectives

By the end of this lesson, you will be able to:

- Describe the components of a blockchain system and how they combine to form a network.
- Articulate the role that the proof of work consensus mechanism plays in keeping the blockchain network synchronized.
- Explain how a difficulty factor influences the proof of work algorithm challenge.
- Integrate a proof of work consensus mechanism into the PyChain blockchain.
- Explain why the validation process is important to the integrity of a blockchain.
- Validate an entire blockchain by comparing the hash values between blocks.

Decentralized Systems

Decentralized Systems

In the first two lessons, Python was used to build a basic blockchain that can securely store financial or other data records.

First Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563

Second Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795

Third Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656

Fourth Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656
4	SHG8757	GHDG746	5664QR6	ADFAFDA

Decentralized Systems

Considerations when opening the blockchain:

01

How should the system be shared?

02

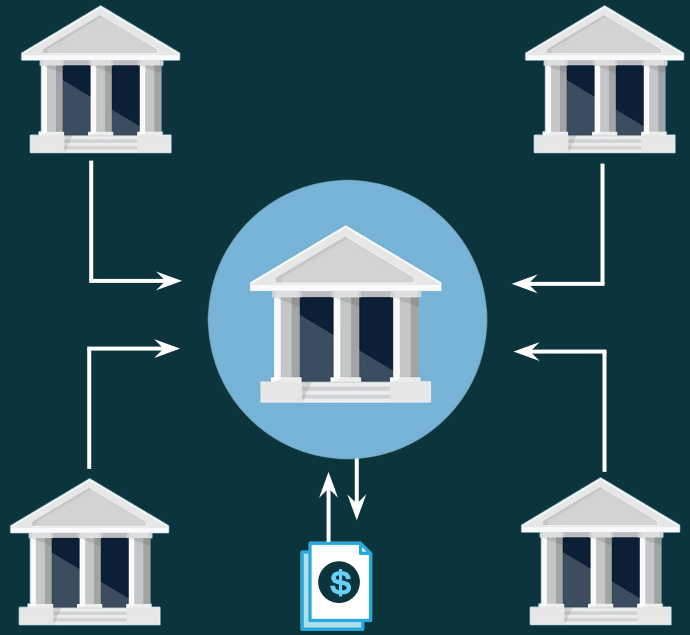
Who should become the central authority over the system and its processes?

03

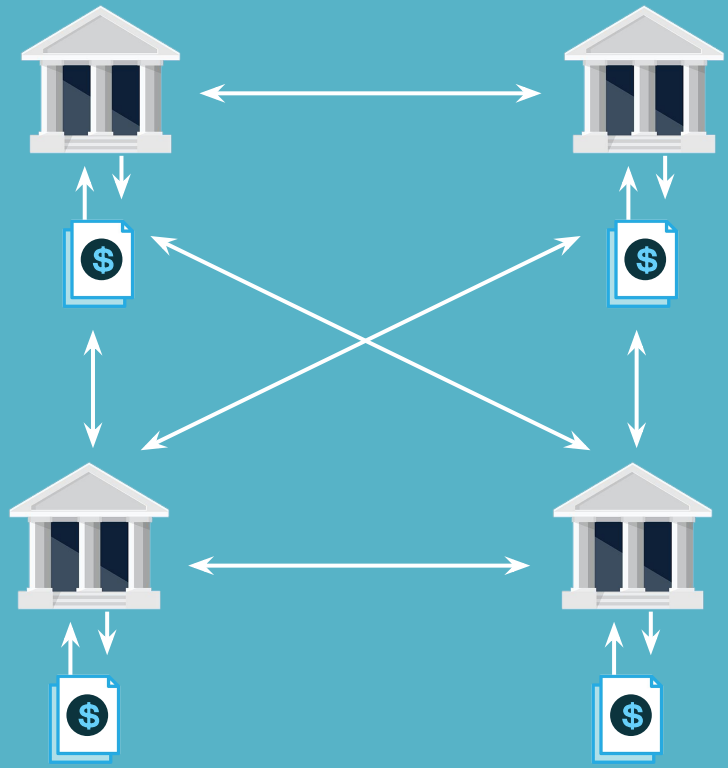
Should a central authority monitor the system in order to ensure the legitimacy of all the transactions? Or, should responsibilities for the ledger be distributed among all of its participants?

Distributing the ledger and delegating responsibility for the system away from a single, central authority is called **decentralization**, which is a key feature of a blockchain.

In a **centralized** system, a central authority monitors and audits transactions in order to protect the system from fraud and other problems.

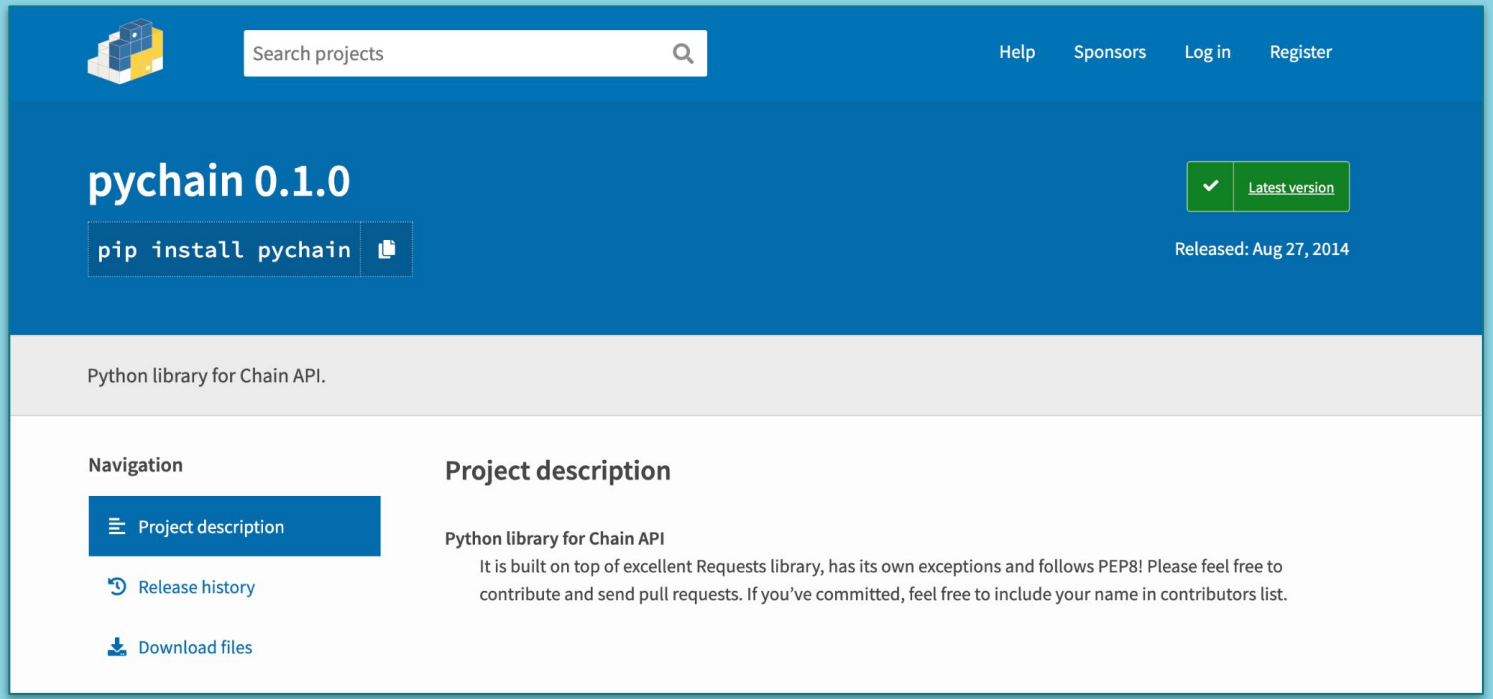


In a **decentralized** blockchain system, no single entity bears the responsibility of monitoring the system. Instead, rules that govern and protect the system are built into the blockchain functionality.



Decentralized Systems

We'll enhance our PyChain by adding an algorithm that allows decentralized participants to agree on recorded transactions across the entire system.

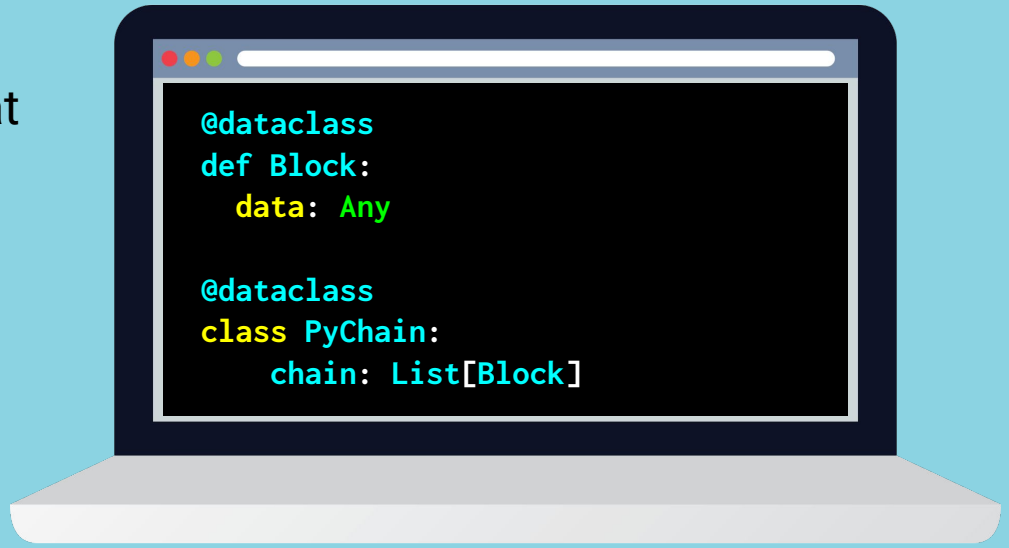


Components of a Blockchain System

Blockchain Nodes

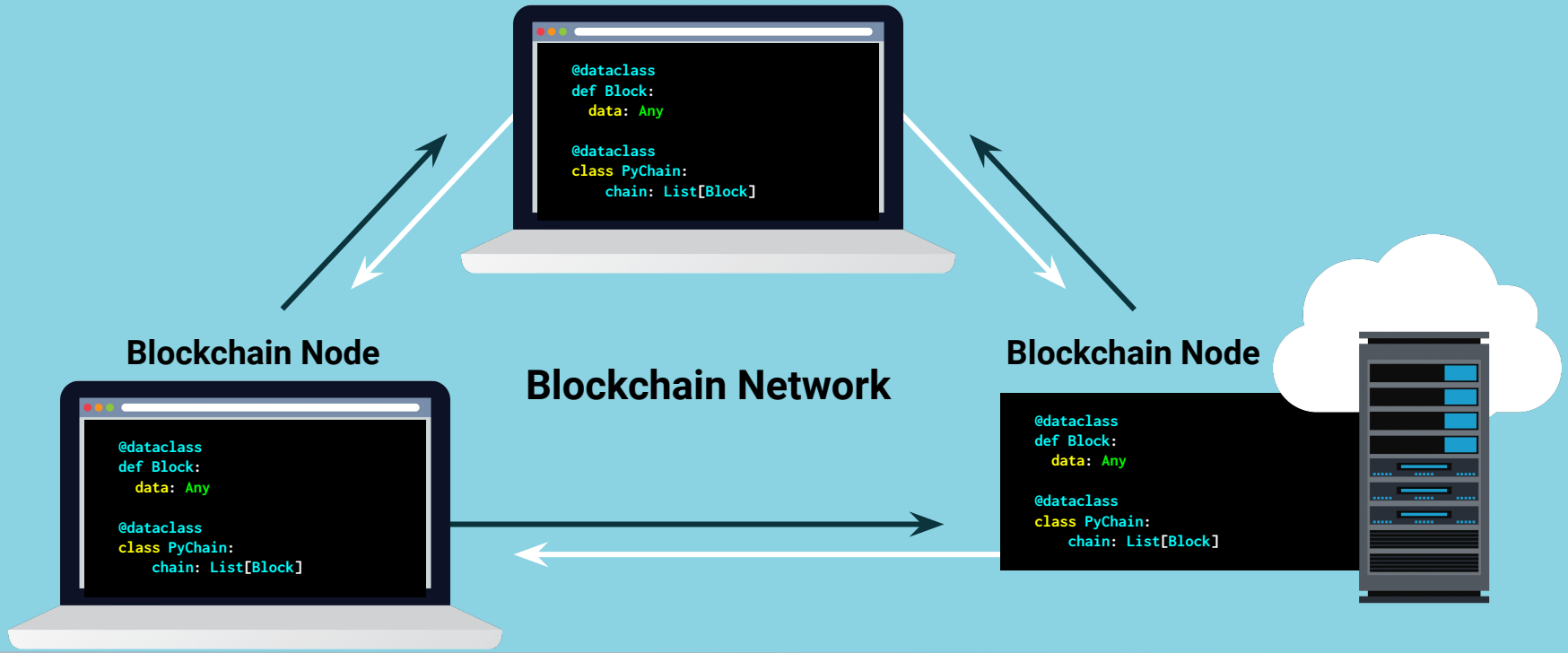
Software, like a Python program, can be used to digitally record data and store it in chained blocks.

In blockchain systems, a device that runs this software—that is, the device that a participant uses to access and add data to the blockchain—is called a node.



Blockchain Nodes

Each node is likely a laptop or desktop computer, but it can also be a server in the cloud.

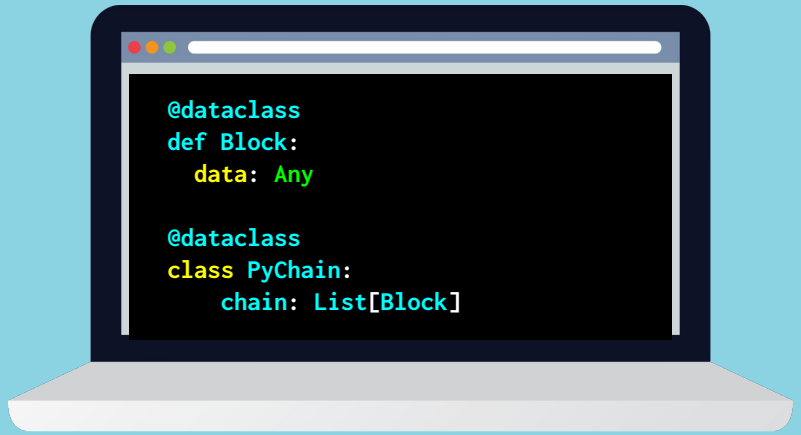


Blockchain Nodes

Each blockchain node has two primary responsibilities:

01 Storing a complete copy of the blockchain.

02 Running the software that establishes the communication and operation of the system.



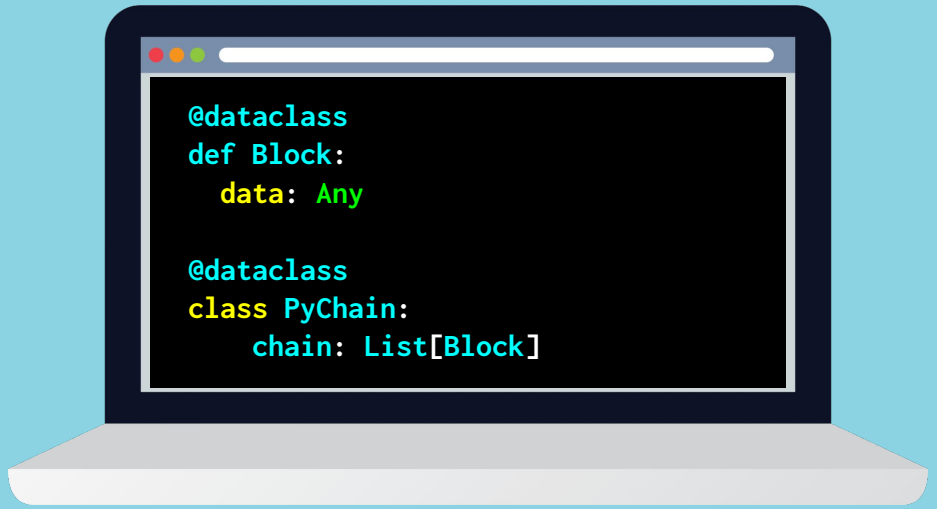
Single-Node System

In a single-node system, the blockchain software is designed to allow users to store and access records through the command line or web application.

Examples:

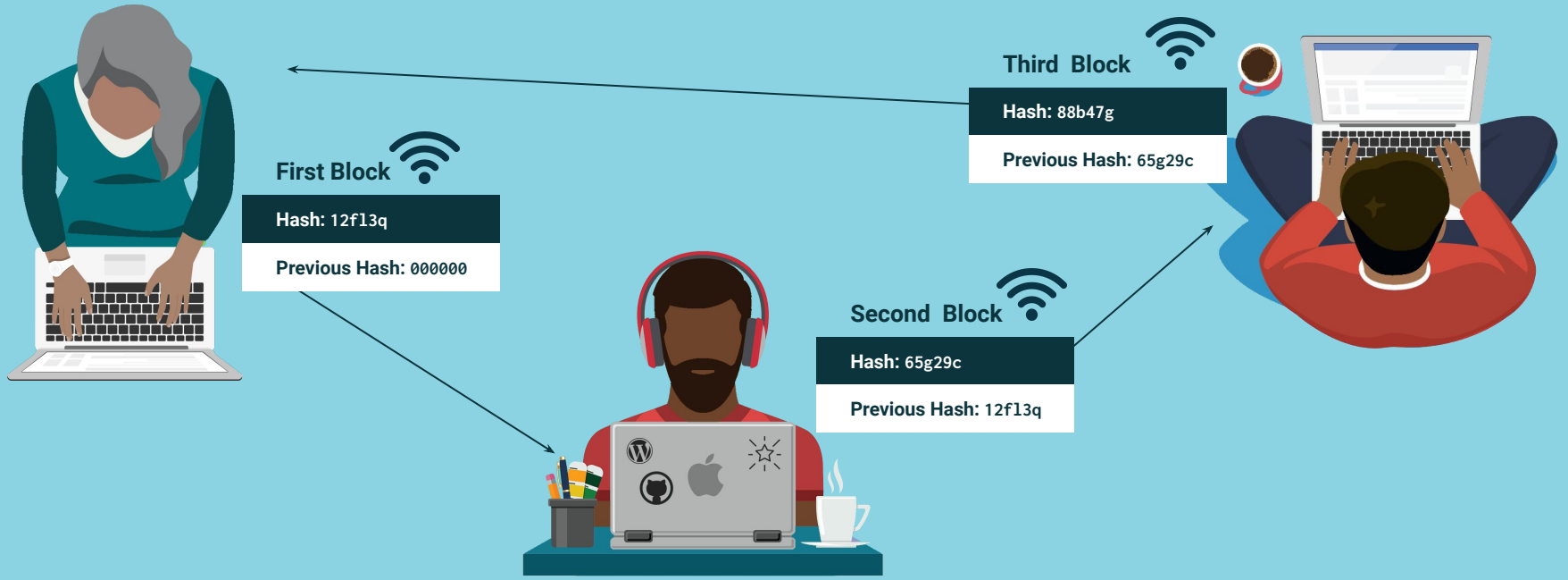
01 A Python program

02 A Streamlit application



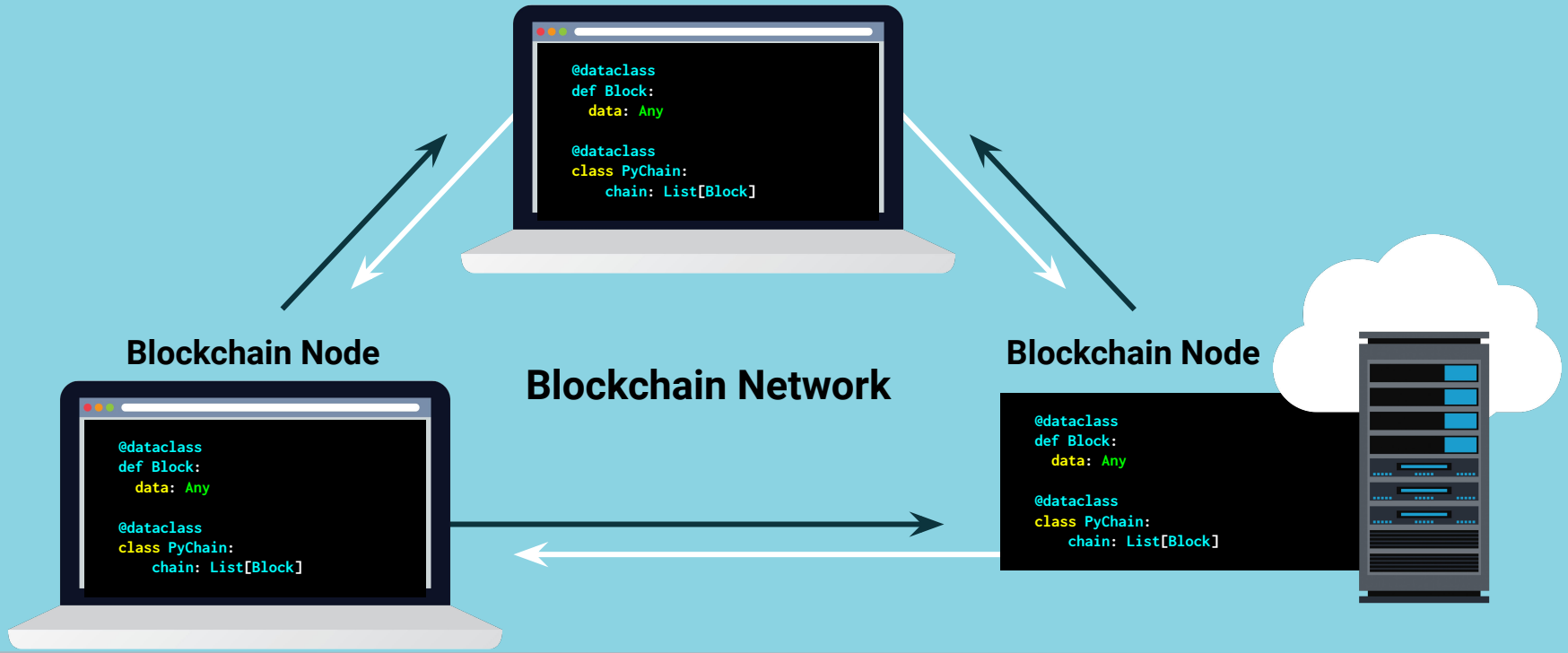
Blockchain Networks

A network is simply a collection of computers that connect to each other in some way. This connection can occur through the internet, over a local WiFi connection, or even via a physical cable that connects two computers.



Blockchain Networks

A blockchain network is composed of all the distributed computers, or nodes, that run a copy of the blockchain.



Blockchain governance: The logic that’s required to maintain the operation, communication, and rules for a blockchain network.

Components of a Blockchain System

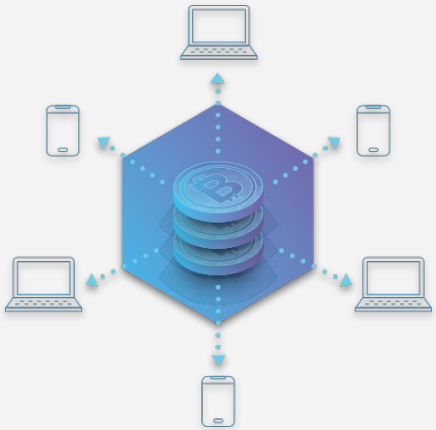
Blockchain governance answers three fundamental questions.

1. What should we do?	The answer determines how copies of the ledger are distributed.
2. Who gets to decide?	The answer determines who gets to add new records to the ledger.
3. How are the deciders chosen and held accountable?	The answer determines how copies of the ledger, maintained on different nodes, are synchronized with each other.

Components of a Blockchain System

Four types of blockchain networks:

Public blockchain



Semiprivate blockchain

Private blockchain

Consortium blockchain

Public and private blockchains are the two most common types.

Public Blockchain

A public blockchain does not restrict access; anyone with an internet connection can send transactions and validate them. Public blockchains are also known as permissionless blockchains.

Block `0`

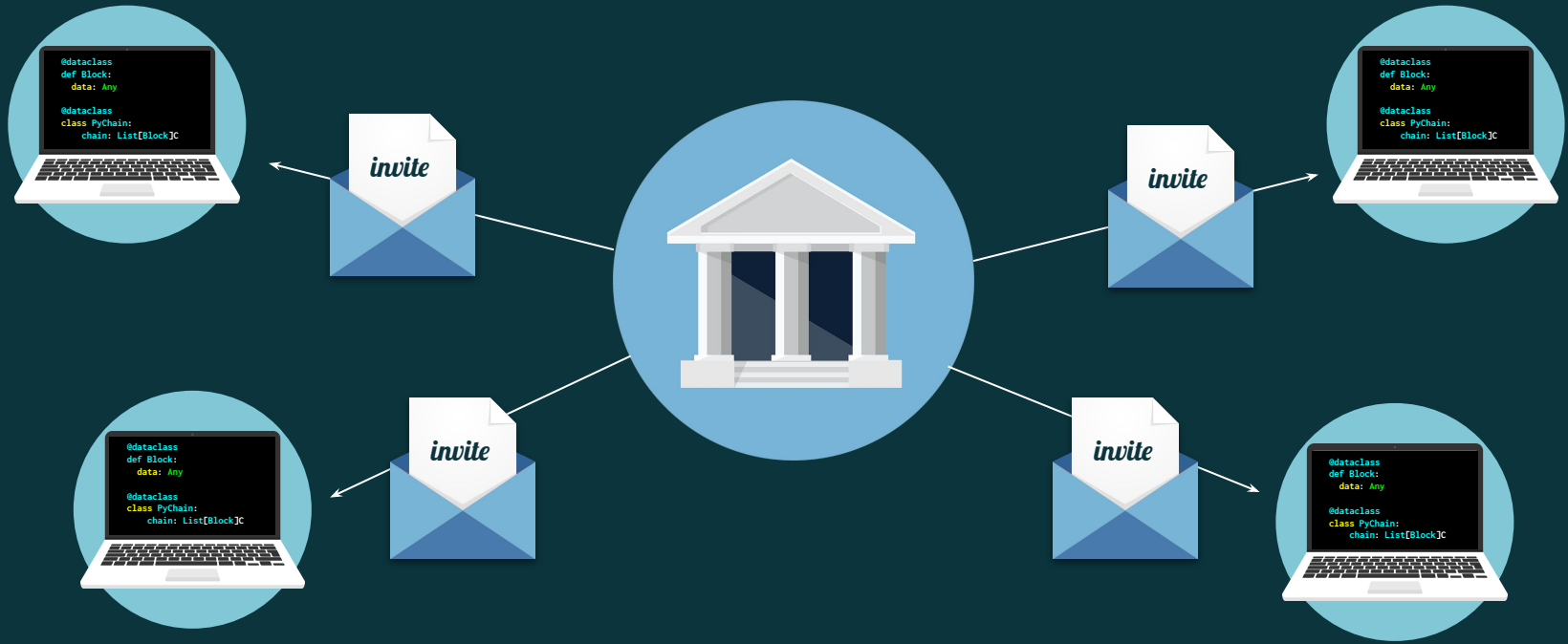
Block 1

The `prev_hash` attribute contains the hash of Block `0`.

Bitcoin and Ethereum are public, permissionless blockchains.

Private Blockchain

In a private blockchain, users must be invited. Private blockchains are also known as permissioned blockchains. Corporate blockchains tend to be private.



A **semiprivate blockchain** combines features of both public and private blockchains in some way.

Consortium Blockchain

A consortium blockchain has restricted access, like a semiprivate blockchain, but consists of two or more groups working together for a common purpose.

Protocol

To communicate with each other, the computers in the network use a protocol. A **protocol** is a set of rules that the members of a communication system use to share information with each other.

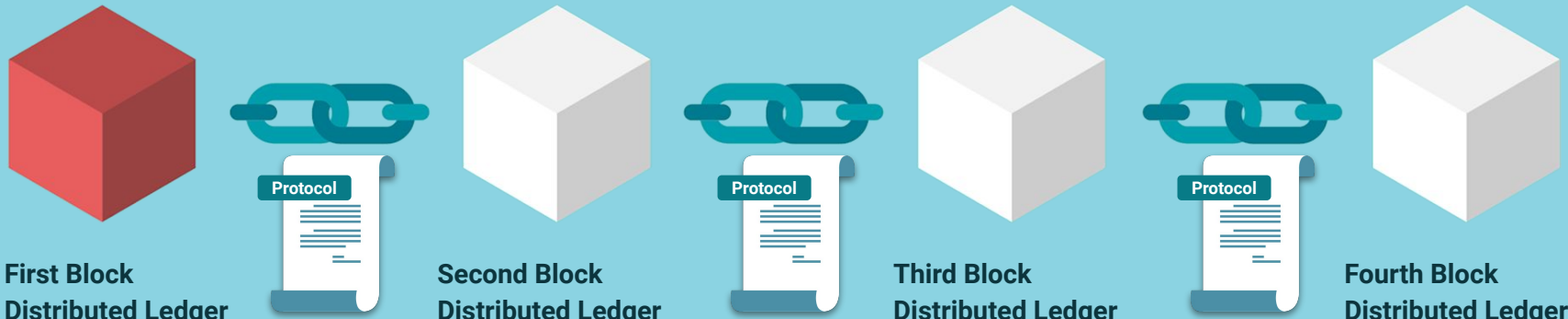
HTTP is a widely used communication protocol.

```
Local URL: http://localhost:8501
Network URL: http://192.168.205.109:8501
```

Ensuring a **consensus** across the decentralized network about the state of the ledger is fundamental to the operation of a blockchain system.

Consensus Protocols

Without consensus, no guarantee would exist that one node's version of the ledger matched any other node's version. Consensus protocols ensure trust in the blockchain system.



First Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563

Second Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795

Third Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656

Fourth Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656
4	SHG8757	GHDG746	5664QR6	ADFAFDA

Popular Consensus Mechanisms

Proof of Work

Though highly effective, proof of work requires a lot of computing and energy resources.

Proof of Stake

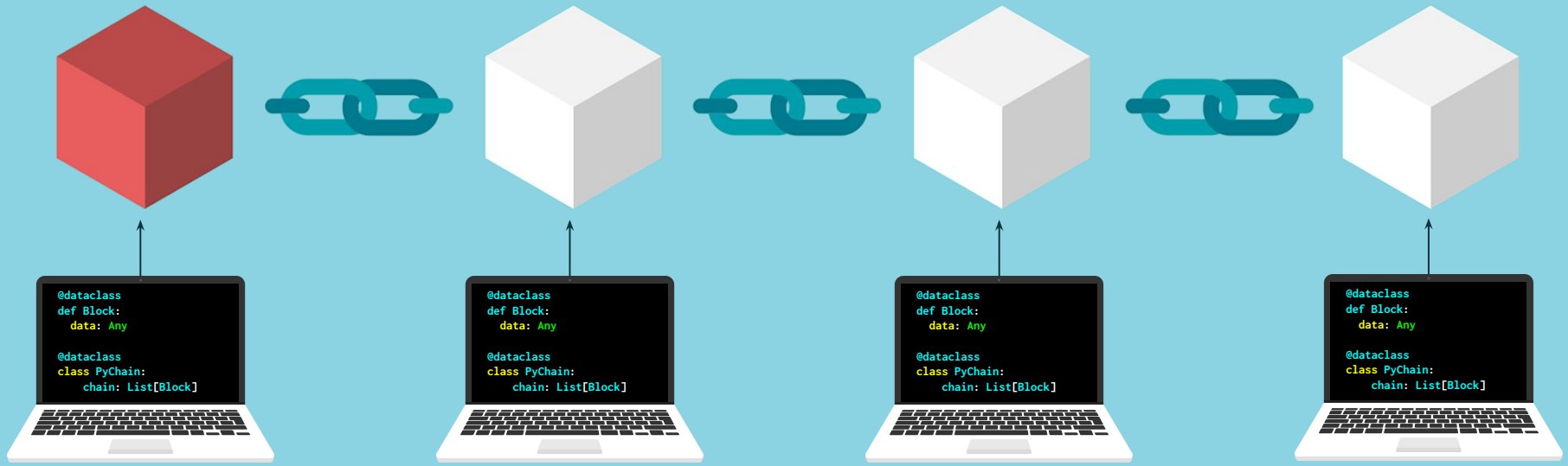
Proof of stake requires less computational energy than proof of work.

Hash Guesser

Proof of Work

A computer can quickly record data to a block and add that block to the chain.

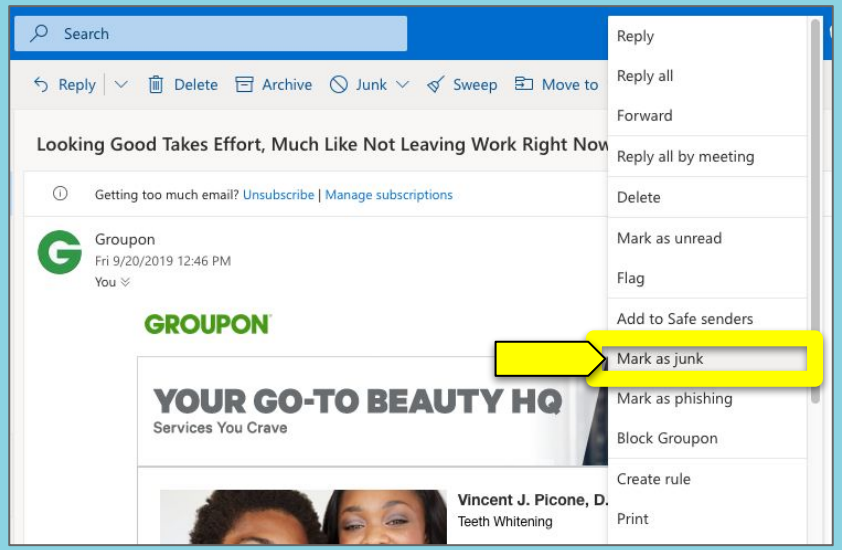
The proof-of-work consensus protocol makes it more difficult to add a block to the chain, which, in turn, makes it harder to cheat the system.



Hashcash Algorithm

One of the most popular consensus algorithms today, proof of work was originally based on the hashcash algorithm, which was created in the 1990s to combat email spam.

```
X-Hashcash: 1:20:1303030600:anni@cypherspace.org::McMybZIhxKXu57jd:ckvi
```



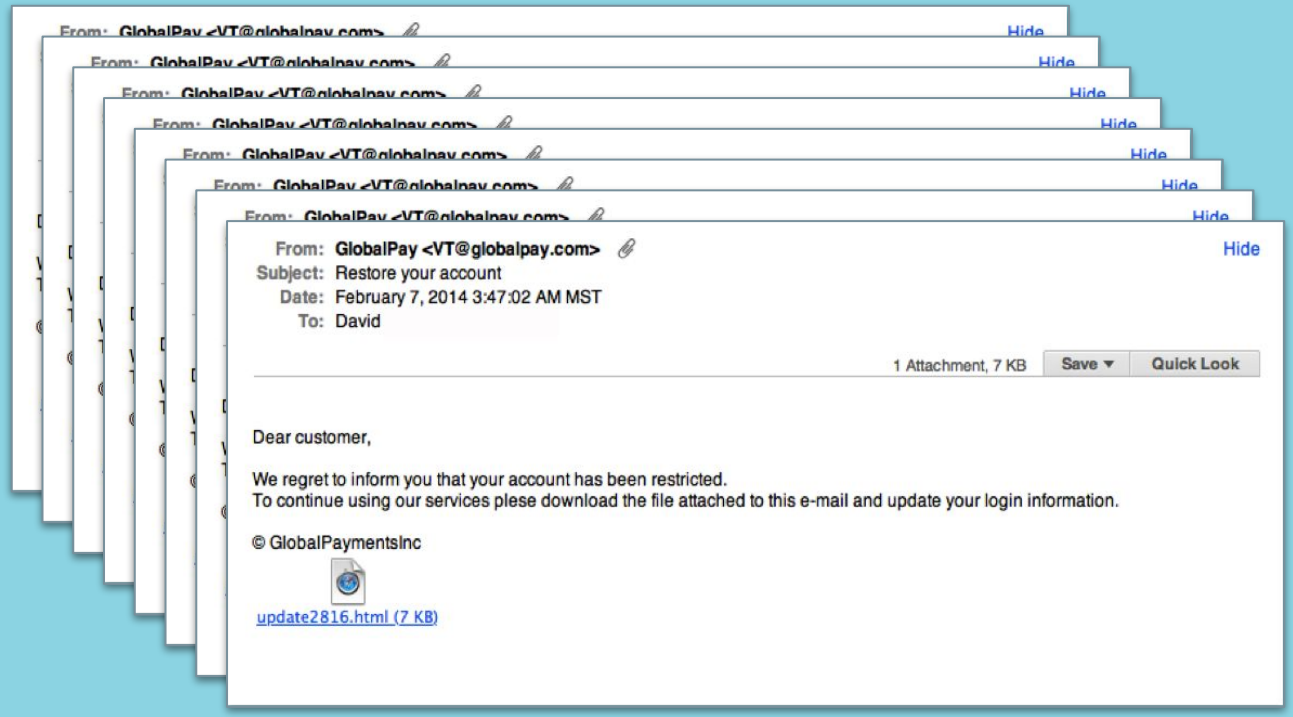
Hashcash Algorithm

The original use case for hashcash was email. Hashcash requires a computer to solve a computational problem before sending an email.

```
X-Hashcash: 1:20:1303030600:anni@cypherspace.org::McMybZIhxKXu57jd:ckvi
```

Hashcash Algorithm

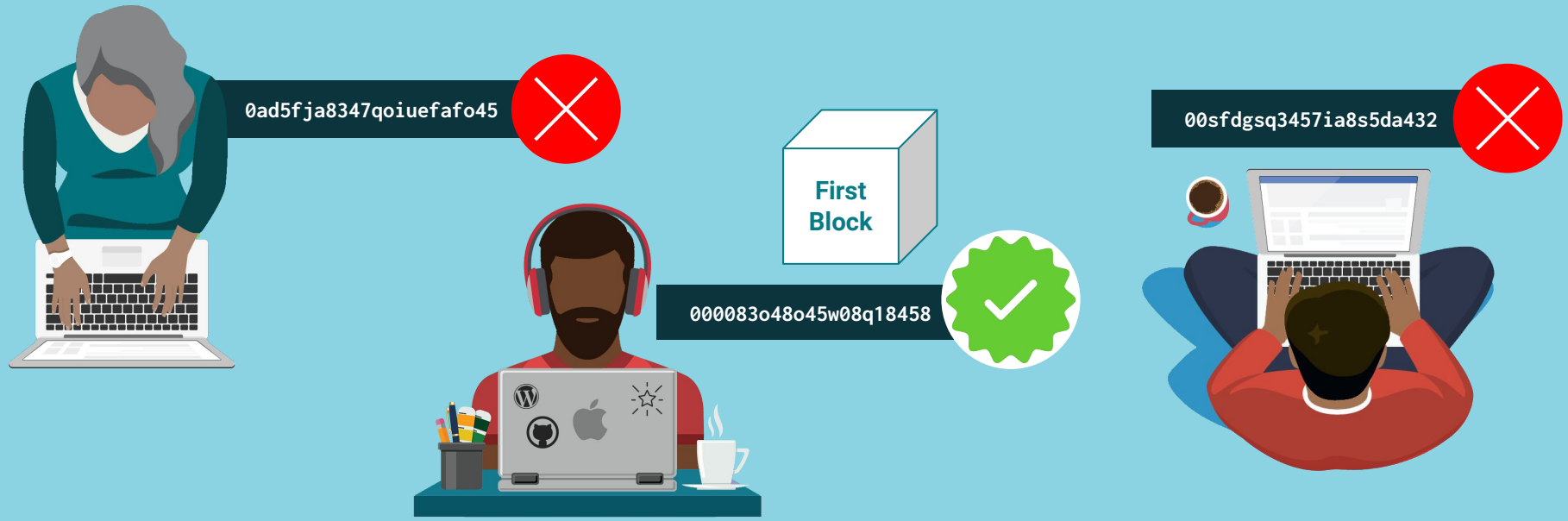
Now, imagine that you're a spammer who wants to send millions of emails every day. This would require a significant of energy, which is expensive.



The hashcash algorithm allows authorized users to use the system without a problem and makes it expensive for malicious users to hack the system. Blockchain applies this same concept.

Hashcash Algorithm

In a blockchain, adding a block to the chain involves a guessing game for the participants. Specifically, the algorithm chooses a random number. Then, each participant guesses a number until someone guesses correctly. The first participant to guess the correct number gets to add a block to the chain.

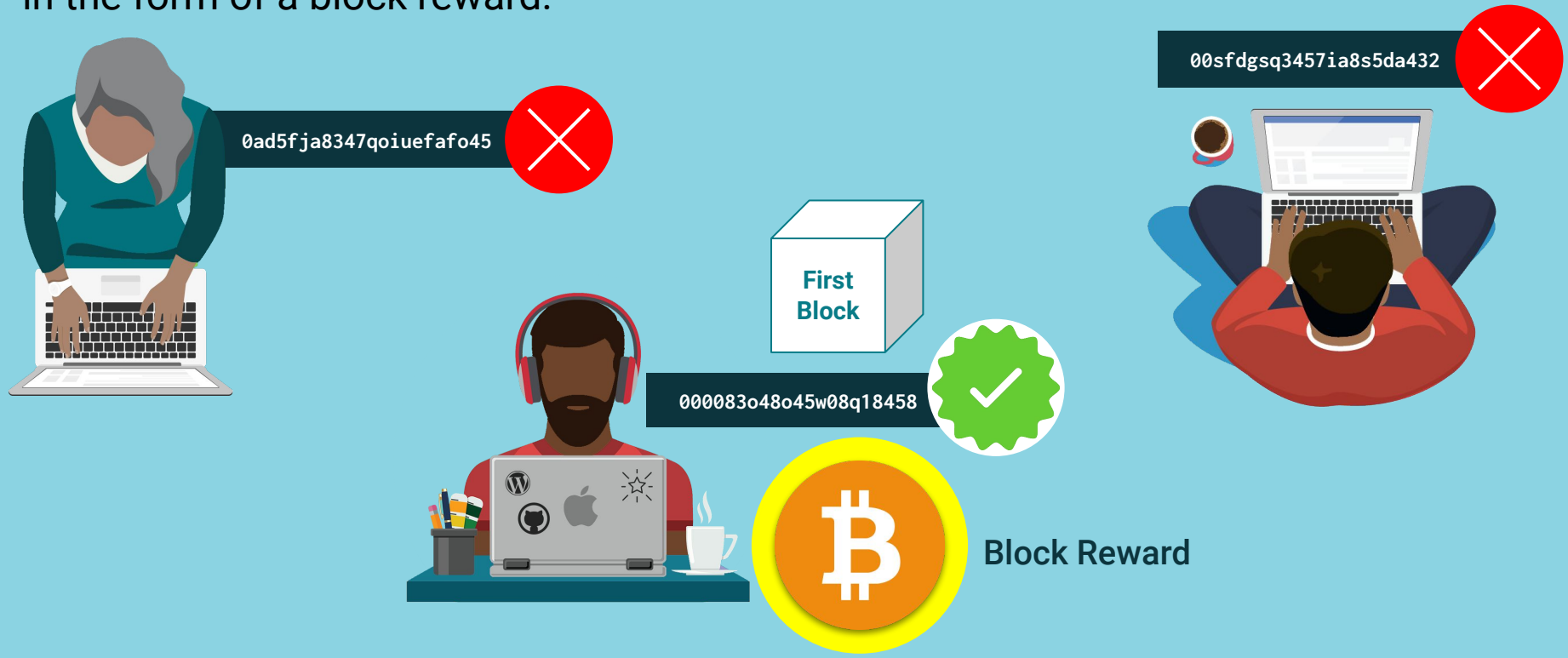


The node that completes the work first wins the privilege of adding a new data block to the chain.

Why is adding a new block to the chain considered a privilege?

Block Reward

Because the person who adds the block receives compensation for their efforts in the form of a block reward.



A **block reward** is a transaction that allocates funds to the user who spent the energy to build that block. The source of these funds varies across blockchains.

Block Reward

Bitcoin

In the Bitcoin blockchain, the funds are allocated by the chain itself.

Ethereum

The block reward on the Ethereum blockchain, is determined by the price the participants in the transaction are willing to pay in fees to the miner.

In blockchain, the process of doing work to add a block to the chain is called **mining**. And, the participants who want to add a new block are called **miners**.

Proof of Work Process

At a high level, the proof of work process is as follows:

01

The system sets the difficulty of the work that's required to add a block to the chain. The difficulty determines how much effort, or computational power, will be required to complete the necessary work.

02

The participants compete to be the first to finish the work.

03

The first to finish gets to add a record to the ledger.

Proof of Work Process

The proof-of-work algorithms that blockchains like Bitcoin and Ethereum require miners to generate a cryptographic hash for the previous block on the chain. The hash contains a specific pattern determined by the algorithm.

Proof of work requires miners to find a set of inputs to include in their blocks that results in a specific pattern in the block hash.

The blockchain system has a pattern that usually requires a certain number of zeros at the beginning of the block hash.

```
0000 3fa996ced47773f2dea29cce9b11f951e6dafe321a84ac7d32791c3b4660
```

Instructor Demonstration

Generate the Correct Hash Pattern

Questions?

Time to Code

Add Hashing to a Block

Suggested Time:

20 minutes

Questions?

Time to Code

Integrate Proof of Work Into PyChain

Suggested Time:

20 minutes

Activity: Dynamic Difficulty

In this activity, you will test the integration of the proof-of-work consensus protocol into the PyChain application.

Suggested Time:

20 minutes

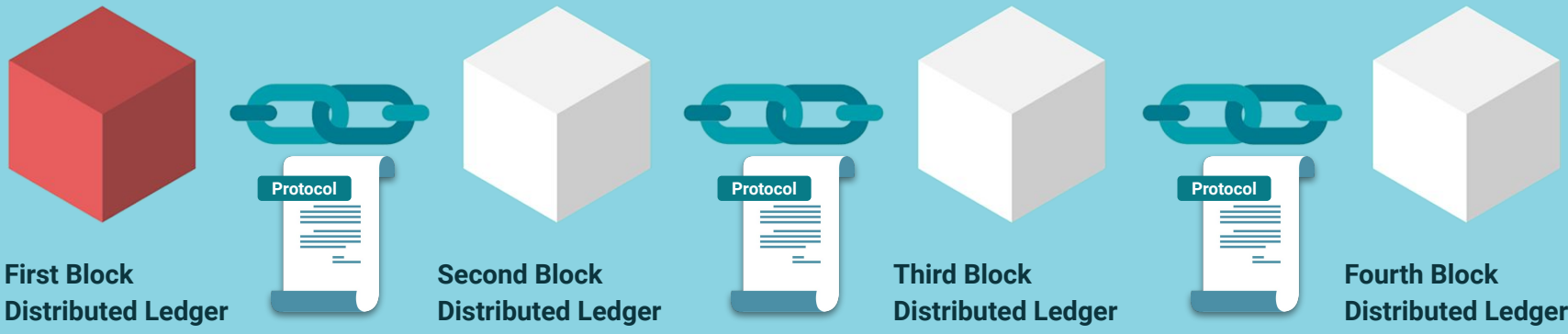
Time's Up! Let's Review.


Break

Proof-of-Work Validation

Proof-of-Work Validation

Part of the goal of the proof-of-work consensus protocol is to prevent nodes from syncing to an invalid copy of the ledger.



First Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563

Second Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795

Third Block
Distributed Ledger

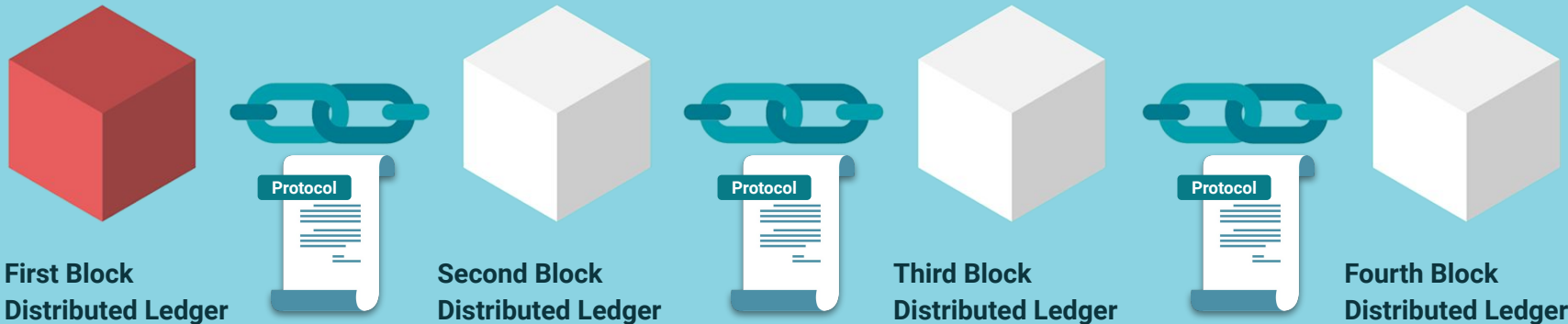
	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656

Fourth Block
Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656
4	SHG8757	GHDG746	5664QR6	ADFAFDA

Proof-of-Work Validation

There's a chance that the miner was a bad actor and altered the state of one of the preceding blocks.



First Block Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563

Second Block Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795

Third Block Distributed Ledger

	A	B		D
1	XC14ADF			6563
2			W4W6	6785795
3	S	7578576	GHD784	L787656

Invalid

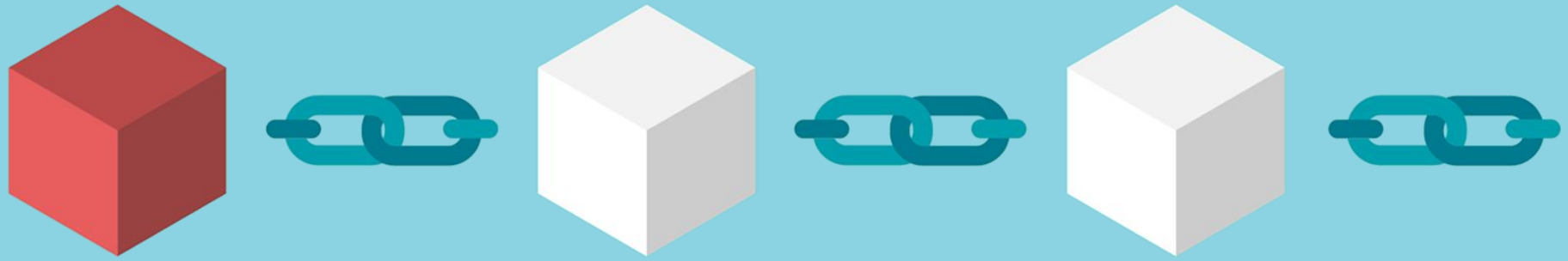
Fourth Block Distributed Ledger

	A	B	C	D
1	XC14ADF	T45W46	A563537	D546563
2	N579689	J788567	Q4W4W6	6785795
3	SFG5354	7578576	GHD784	L787656
4	SHG8757	GHDG746	5664QR6	ADFAFDA

Or, the miner could claim to have found the right nonce value to generate a hash that matches the requirement, but in reality, they have not done the work.

Proof-of-Work Validation

The blockchain is validated by adding cryptographic links between all the blocks in the ledger through their hashes.



First Block

Data: `"Good Day"`

Previous Hash: `000000`

Hash: `12fl3q`

Second Block

Data: `"Thursday"`

Previous Hash: `12fl3q`

Hash: `d340kc`

Third Block

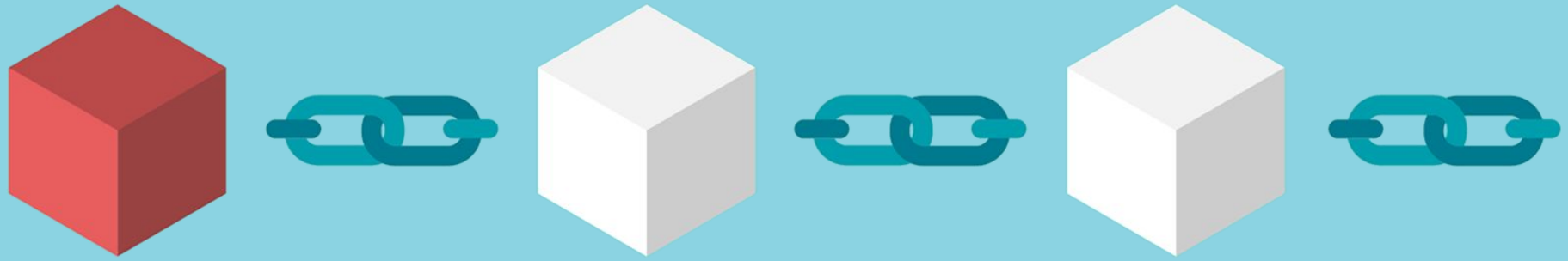
Data: `"Fingerprint"`

Previous Hash: `d340kc`

Hash: `lm12ep`

Proof-of-Work Validation

It's possible to manually validate the entire chain by calculating the hash of a block, and then checking if the `prev_hash` attribute value in the next block matches the current block's hash.



First Block

Data: `"Good Day"`

Previous Hash: `000000`

Hash: `12fl3q`

Second Block

Data: `"Thursday"`

Previous Hash: `12fl3q`

Hash: `d340kc`

Third Block

Data: `"Fingerprint"`

Previous Hash: `d340kc`

Hash: `lm12ep`

Proof-of-Work Validation

Steps for validating the PyChain ledger:

01 Hash the first block in the chain.

02 Access the `prev_hash` attribute value from the next block.

03 Compare the two hashes.

04 Repeat Steps 1 through 3 for each block in the chain until every block's `prev_hash` value has been evaluated.

If all comparisons result in matches, the entire blockchain is valid. Otherwise, it's invalid.

Instructor Demonstration

Validate the Chain

Questions?

Activity: Validating the Blockchain

In this activity, you will use Streamlit to test the validation functionality associated with the PyChain.

Suggested Time:

20 minutes

Time's Up! Let's Review.


The End