

# Chain Verse

## Understanding Blockchain Architecture & Web3 (Session - 1 : Theory)



**Mrs. Lifna C S**

**Assistant Professor**

**Department of Computer Engineering**

**Vivekanand Education Society's Institute of Technology, Mumbai**

Dated : 26th Feb 2026

1. Introduction to Blockchain
2. Blockchain Fundamentals
3. Cryptocurrency Basics
4. Introduction to Solidity Programming

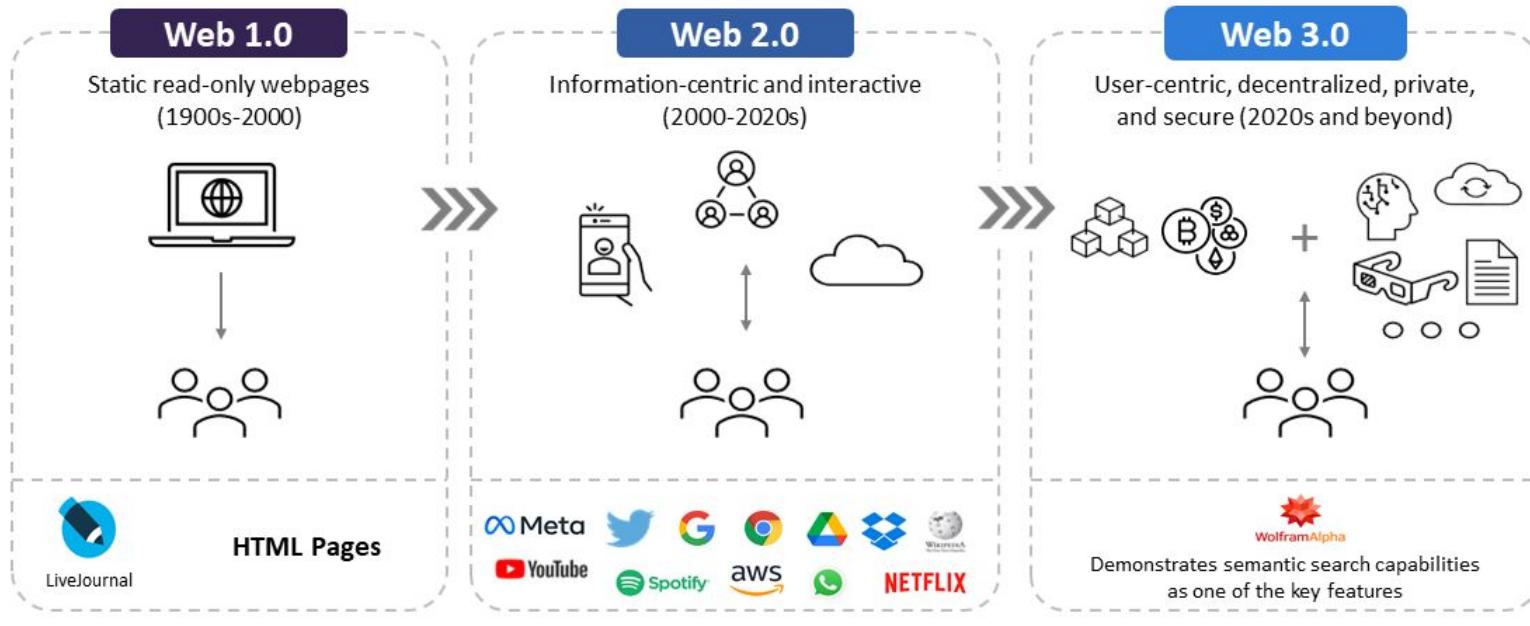


# 1. Introduction to Blockchain

## Evolution of Web



**Web 3.0 is the evolution of the internet towards user-centric intelligent services**



# 1. Introduction to Blockchain

## Why to learn Blockchain ?

### Current Scenario

- Internet is owned by Technical Giants
- Huge Transaction fees by 3rd Parties
- Time to complete Transactions..
- Ownership for Content Creators
- Lack of Transparency

### Blockchain Offers ...

- Decentralized with P2P Network
- Trust in a Trustless Network
- Immutable
- Security through Cryptography
- Transparency





# 1. Introduction to Blockchain



## What is Web3?

- Decentralized version of the internet
- Powered by blockchain and smart contracts
- User-owned data and digital assets

## Benefits of Web3

- User ownership and control
- Censorship resistance
- Trustless transactions
- Global accessibility

## Key Components of Web3

- Blockchain platforms (Ethereum, Polygon)
- Smart Contracts
- Decentralized Storage (IPFS)
- Tokens and Cryptography

## Real-World Use Cases

- Decentralized Finance (DeFi)
- NFTs and Digital Ownership
- DAOs and Governance
- Supply Chain & Healthcare

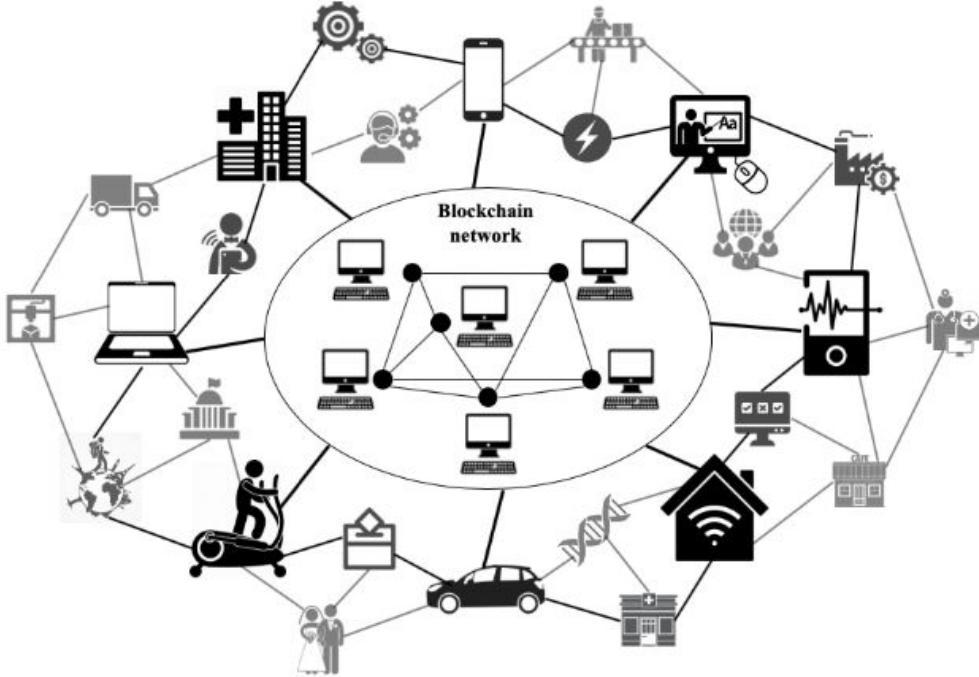
# 1. Introduction to Blockchain

## What is Decentralization?

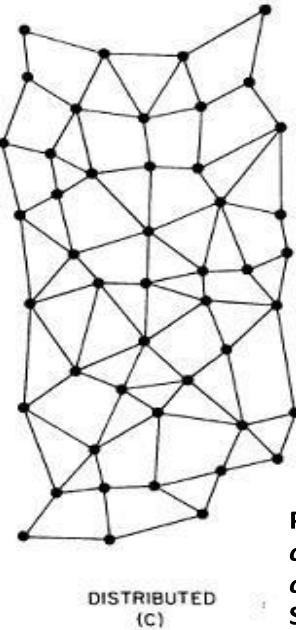
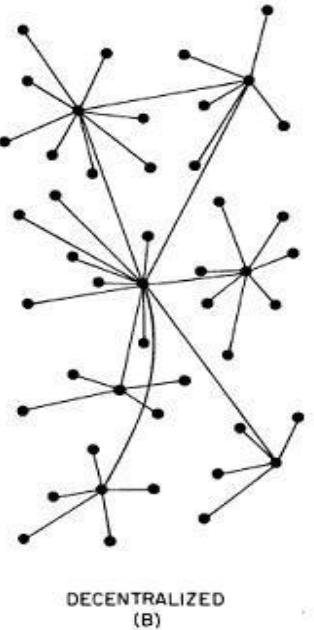
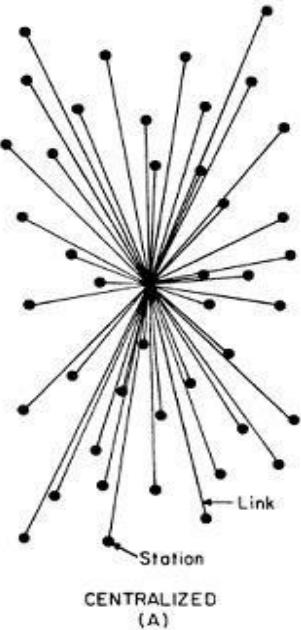
- No single controlling authority
- Distributed nodes and governance
- Improved resilience and transparency

## Backbone of Blockchain

- Distributed ledger technology
- Consensus-based validation
- Immutable and transparent records



## Centralized vs Decentralized vs Distributed



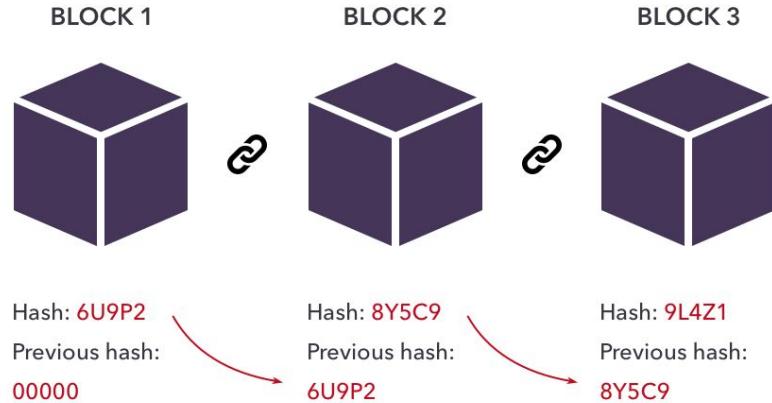
Complete reliance on single point (**centralized**) is not safe

- **Decentralized:** Multiple points of coordination
- **Distributed:** Everyone collectively execute the job

Photo courtesy: Baran, Paul. *On distributed communications: I. Introduction to distributed communications networks*. No. RM3420PR. RAND CORP SANTA MONICA CALIF, 1964.

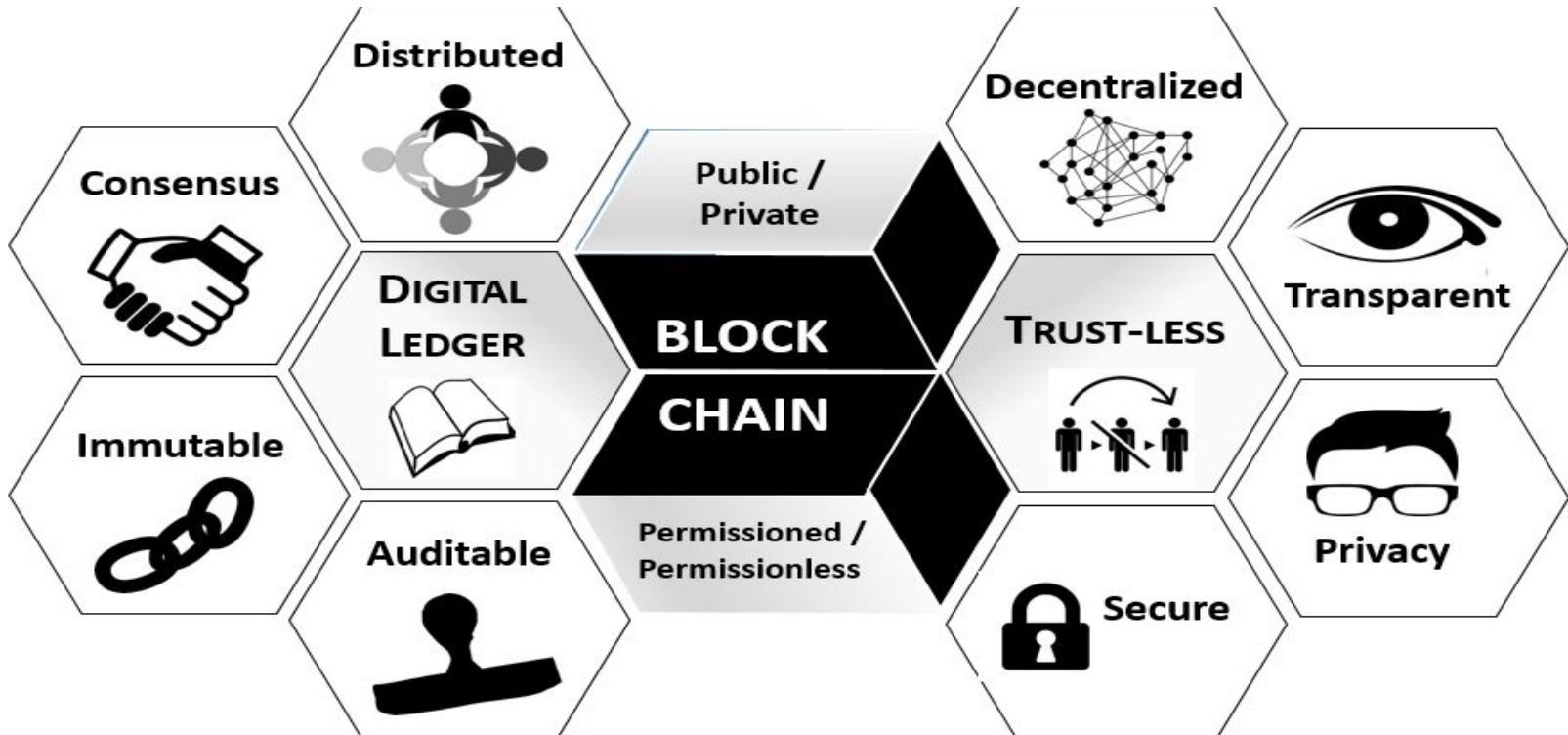
# 1. Introduction to Blockchain

- ✓ Blockchain is a digital public ledger that records online transactions.
- ✓ Blockchain ensures confidentiality, integrity and privacy.
- ✓ When a new block is added to a blockchain, it is linked to the previous block using a cryptographic hash.
- ✓ This ensures the chain is never broken and that each block is permanently recorded.
- ✓ It is also intentionally difficult to alter past transactions in blockchain since all the subsequent blocks must be altered first.



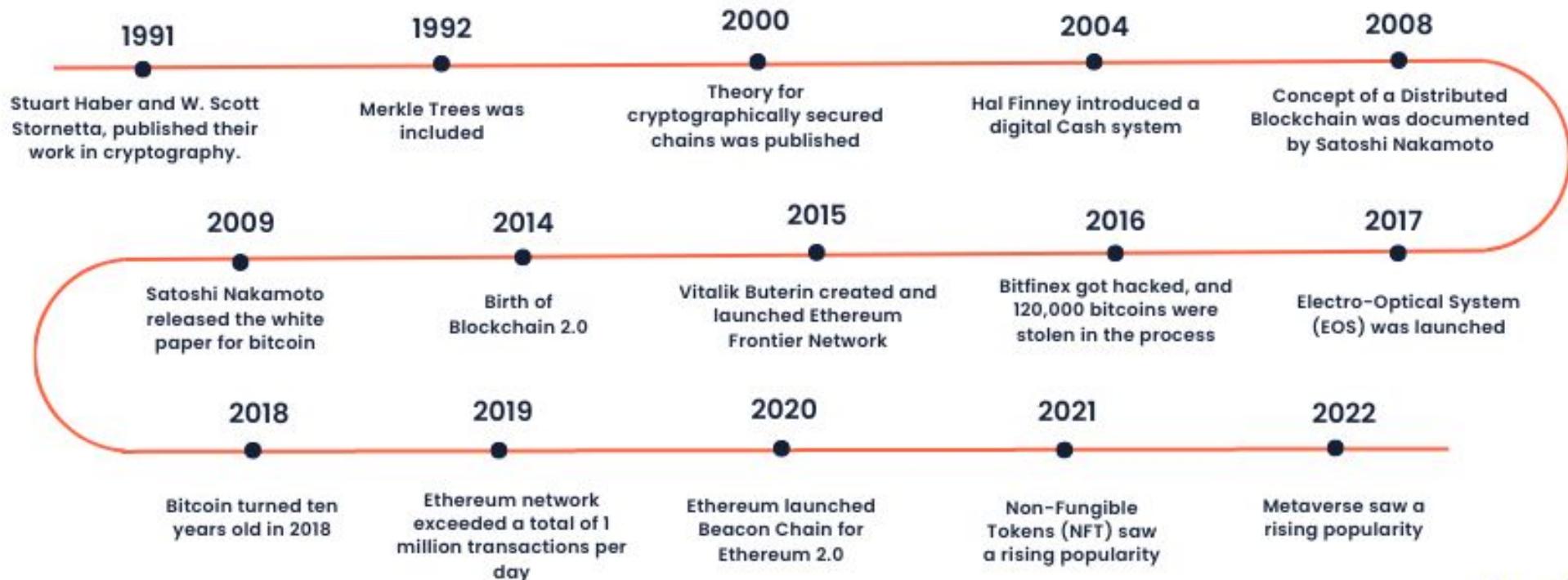
# 1. Introduction to Blockchain

## Characteristics of Blockchain



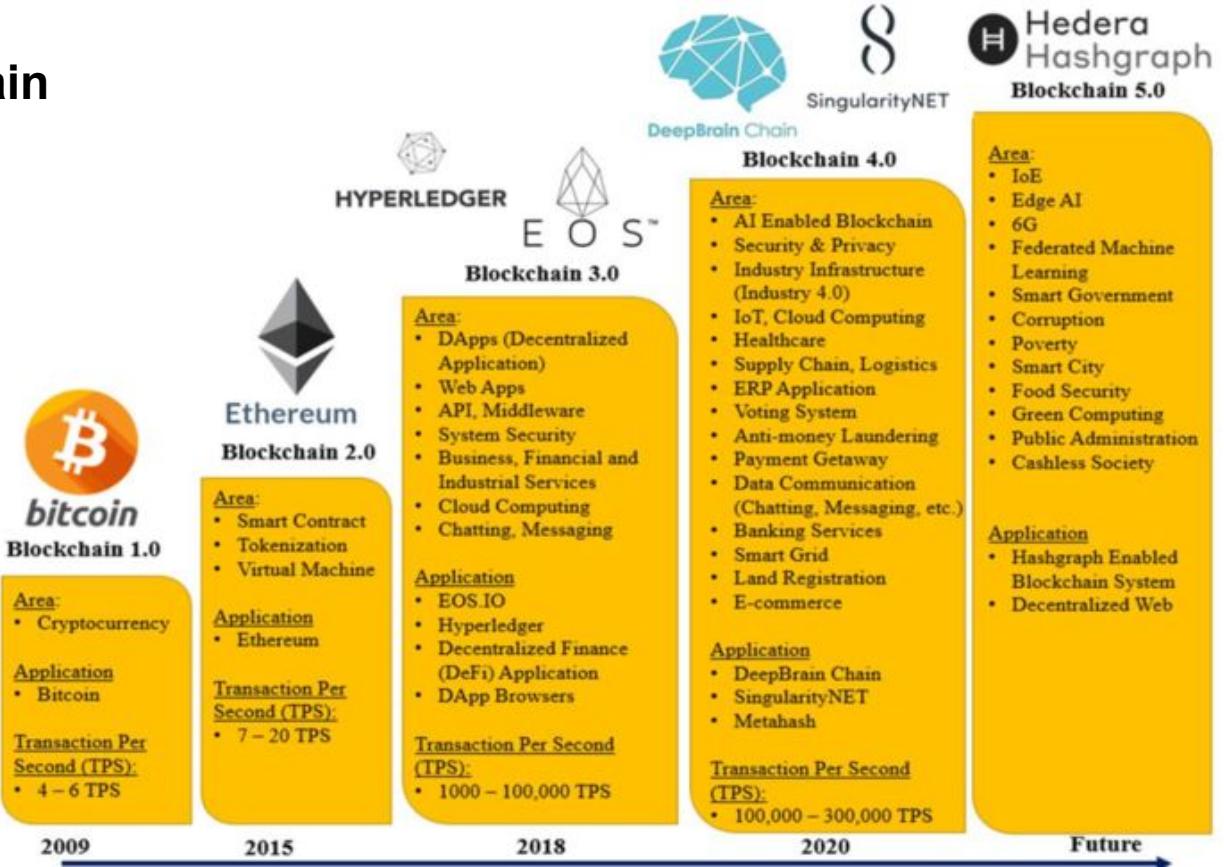
# 1. Introduction to Blockchain

## Blockchain History Timeline



# 1. Introduction to Blockchain

## Generations of Blockchain

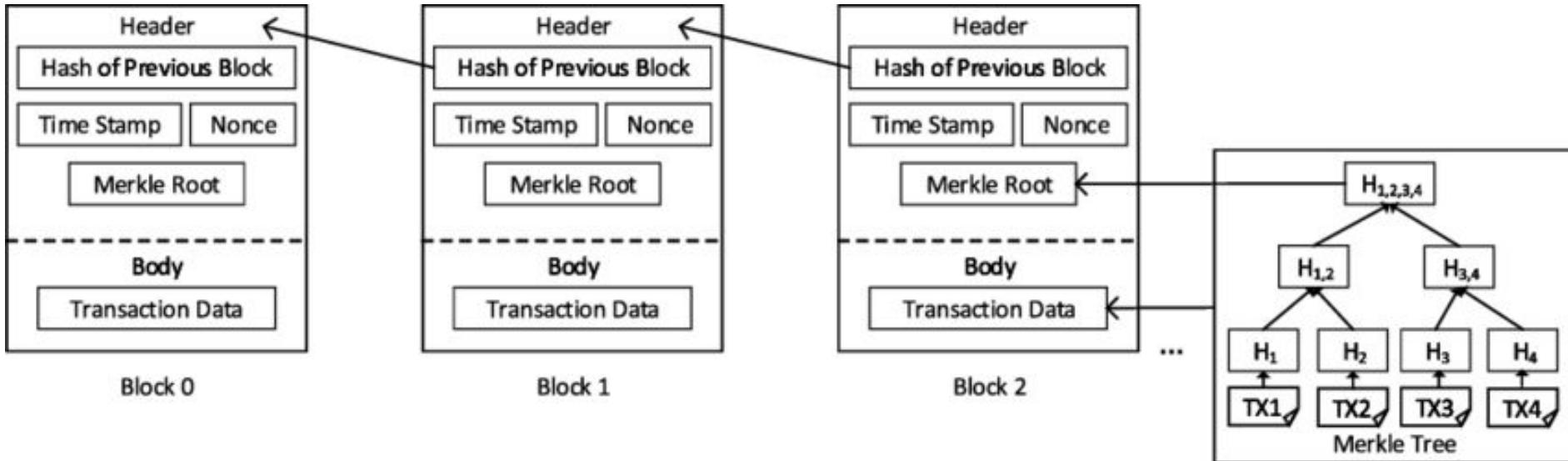


1. Introduction to Blockchain
2. **Blockchain Fundamentals**
3. Cryptocurrency Basics
4. Introduction to Solidity Programming



## 2. Blockchain Fundamentals

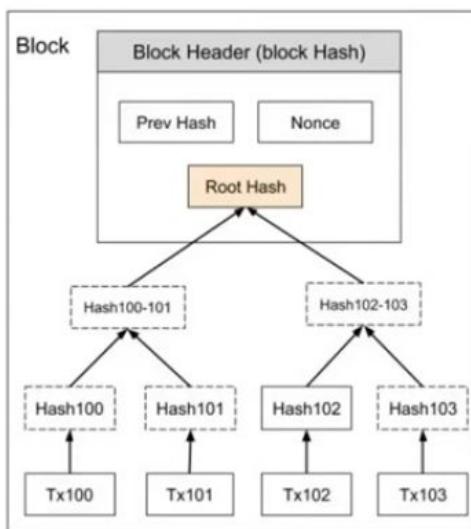
### Blockchain Structure



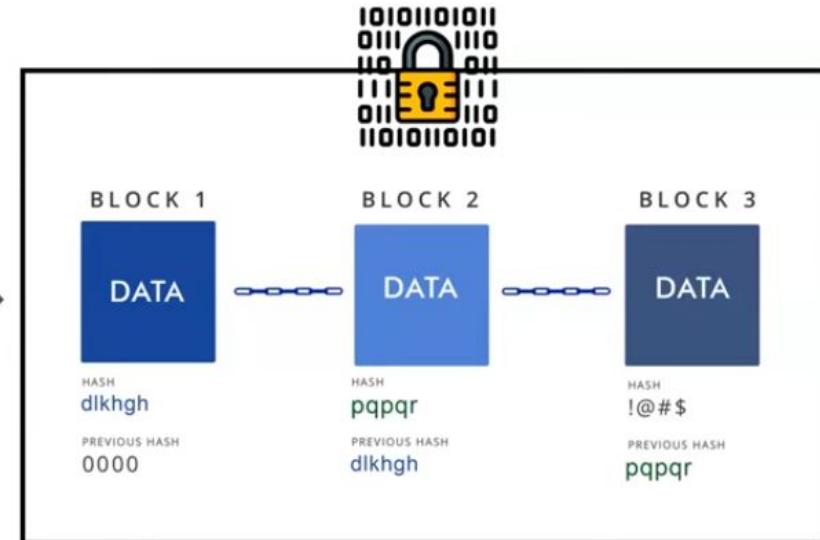
## 2. Blockchain Fundamentals

### Merkle Tree

Merkle trees are a type of data structure commonly used in computer science. They are used to encrypt blockchain data more effectively and securely in bitcoin and other cryptocurrencies.



Merkle Tree

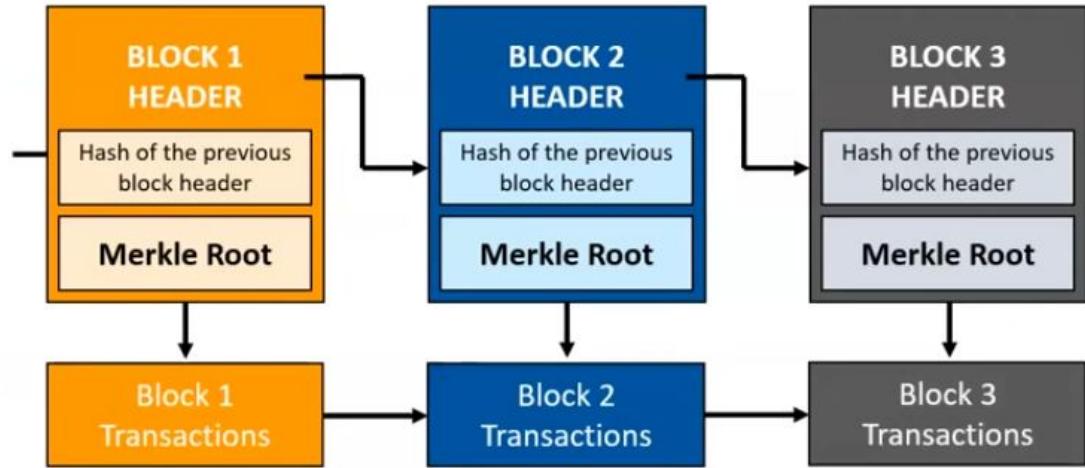


Blockchain Data encrypted Securely

## 2. Blockchain Fundamentals

### Why is it essential for Blockchain ?

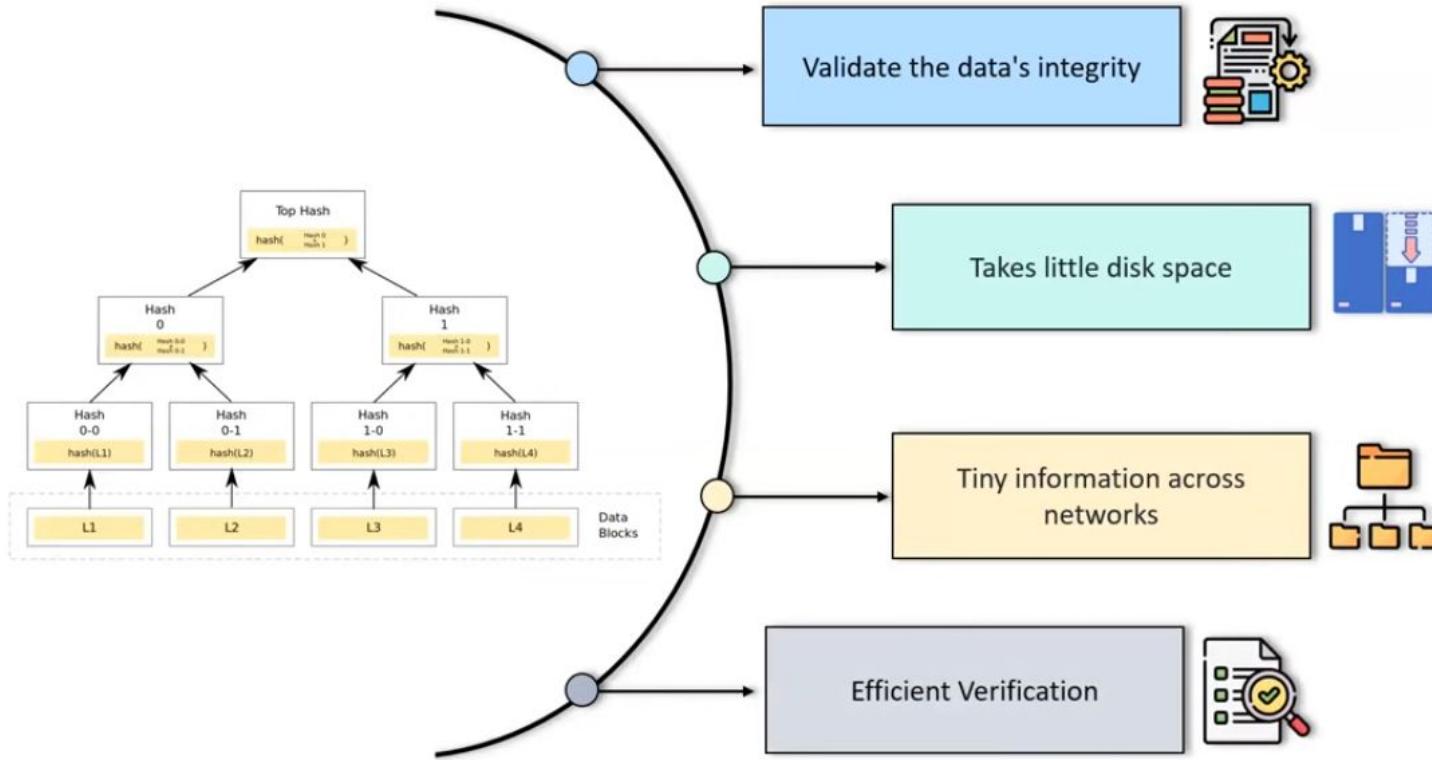
Merkle Trees hash records in accounting, thereby separating the proof of data. Proving that given information across the network is all that is required for a transaction to be valid.



Merkle Tree breaking the data into tiny parts of information

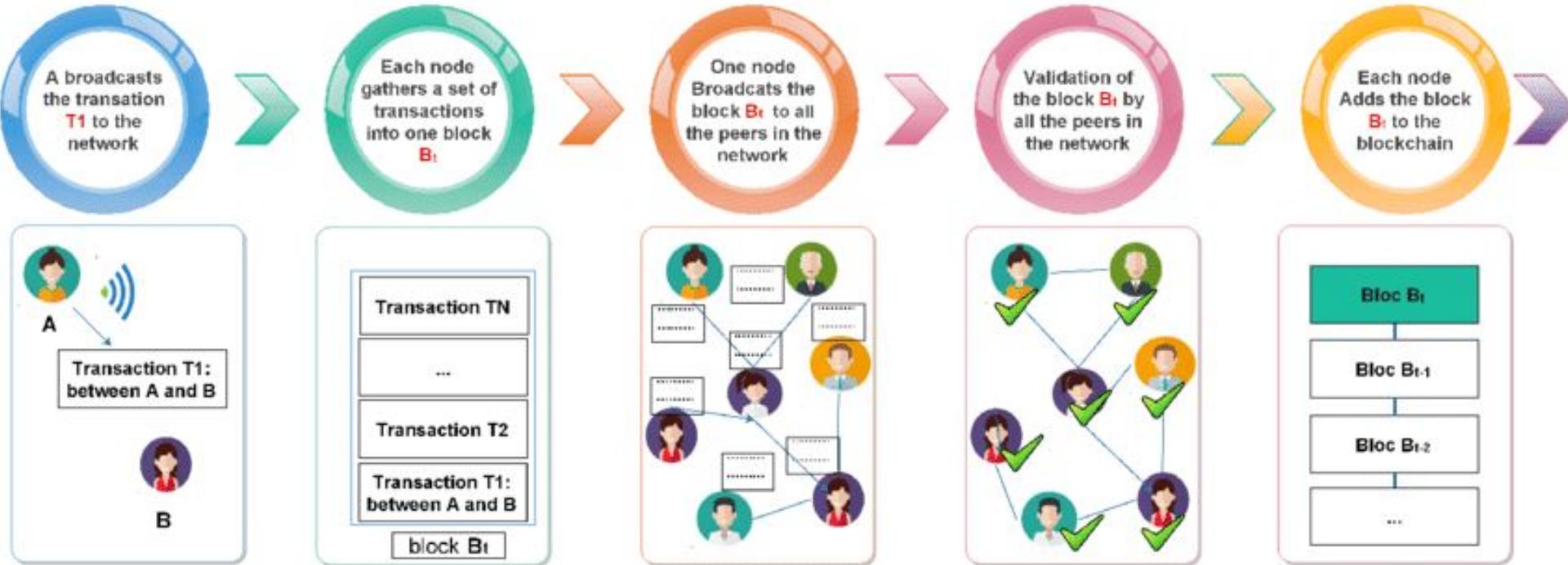
## 2. Blockchain Fundamentals

### Benefits of Merkle Tree



## 2. Blockchain Fundamentals

### Transactions in Blockchain - Example



### WHAT ARE NODES?

Nodes are the electronic devices connected to the network and possessing an IP address. Generally, nodes are the communication endpoints which means that any user or application that wants to interact with the blockchain does so through nodes.

### WHAT DOES A BLOCKCHAIN NODE DO?



Processing A Transaction



Managing the transactions  
and their validity

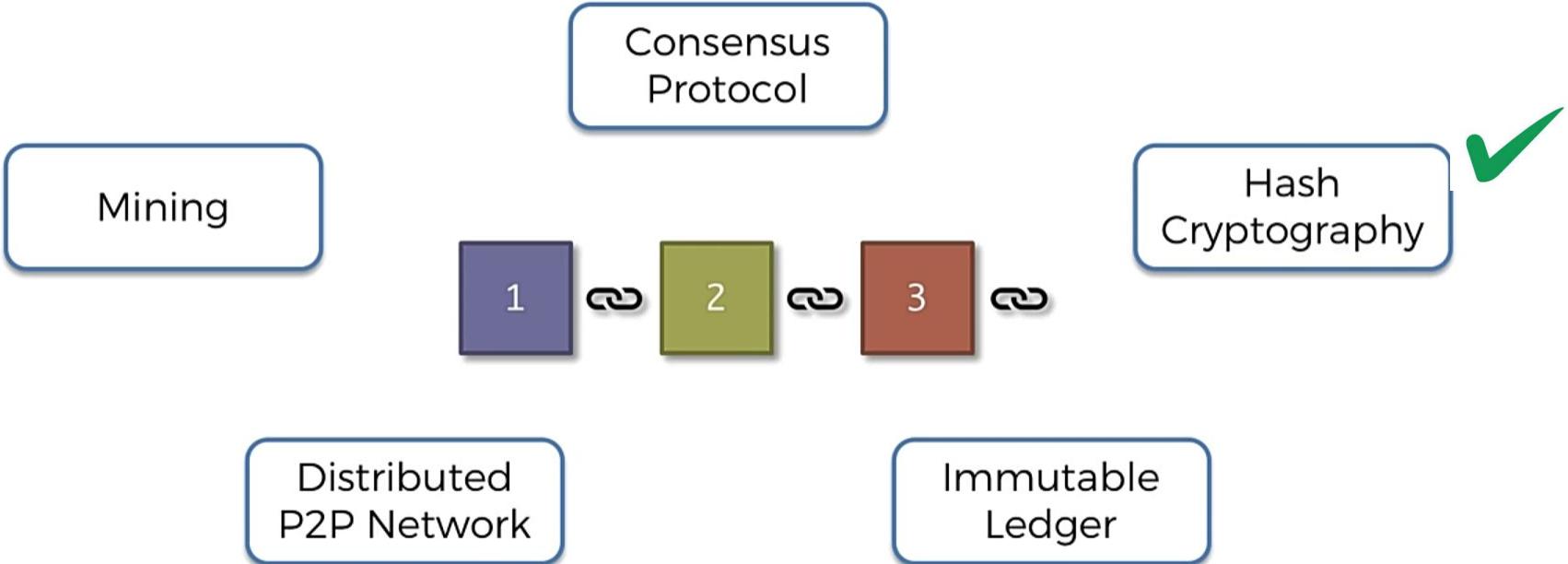


Storing the cryptographically  
linked blocks



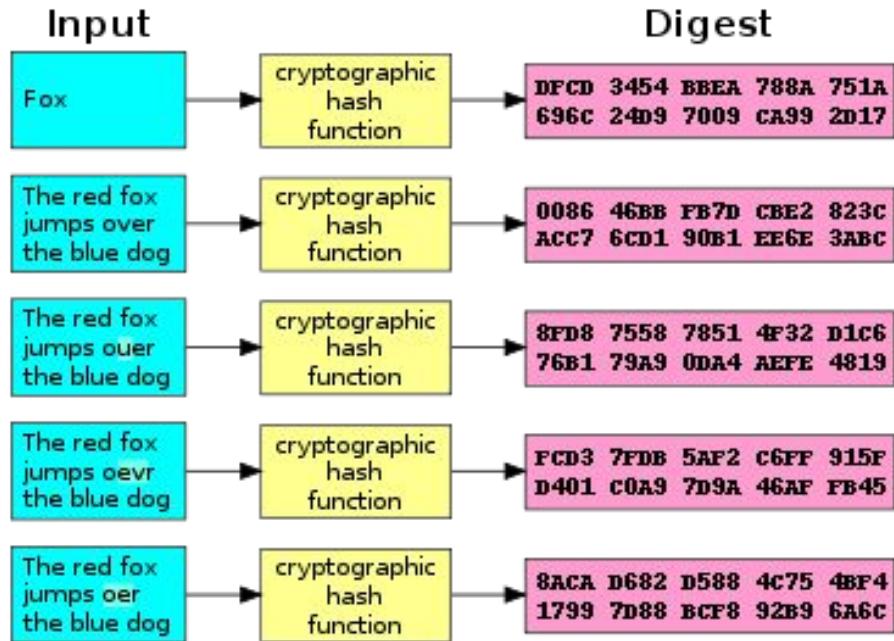
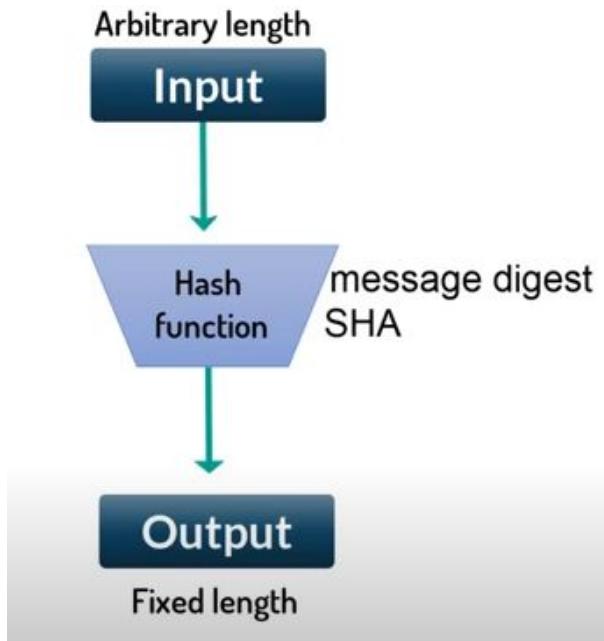
Acting as a point of  
communication

## 2. Blockchain Fundamentals



## 2. Blockchain Fundamentals

### Cryptographic Hash Functions





## 2. Blockchain Fundamentals



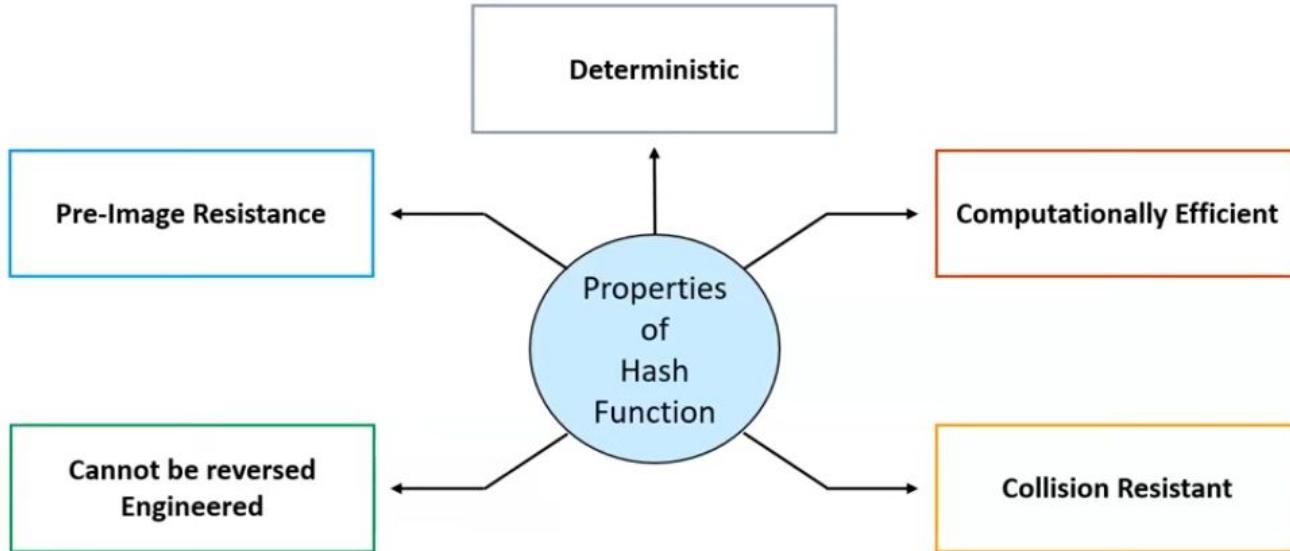
### Cryptographic Hash Functions - Demo

The screenshot shows a web browser window with the URL <https://andersbrownworth.com/blockchain/hash> in the address bar. The page title is "Blockchain Demo". Below the title, there is a navigation bar with tabs: Hash (which is active), Block, Blockchain, Distributed, Tokens, and Coinbase. The main content area is titled "SHA256 Hash". It contains two input fields: one labeled "Data:" and another labeled "Hash:". The "Data:" field is empty, and the "Hash:" field contains the value "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855".

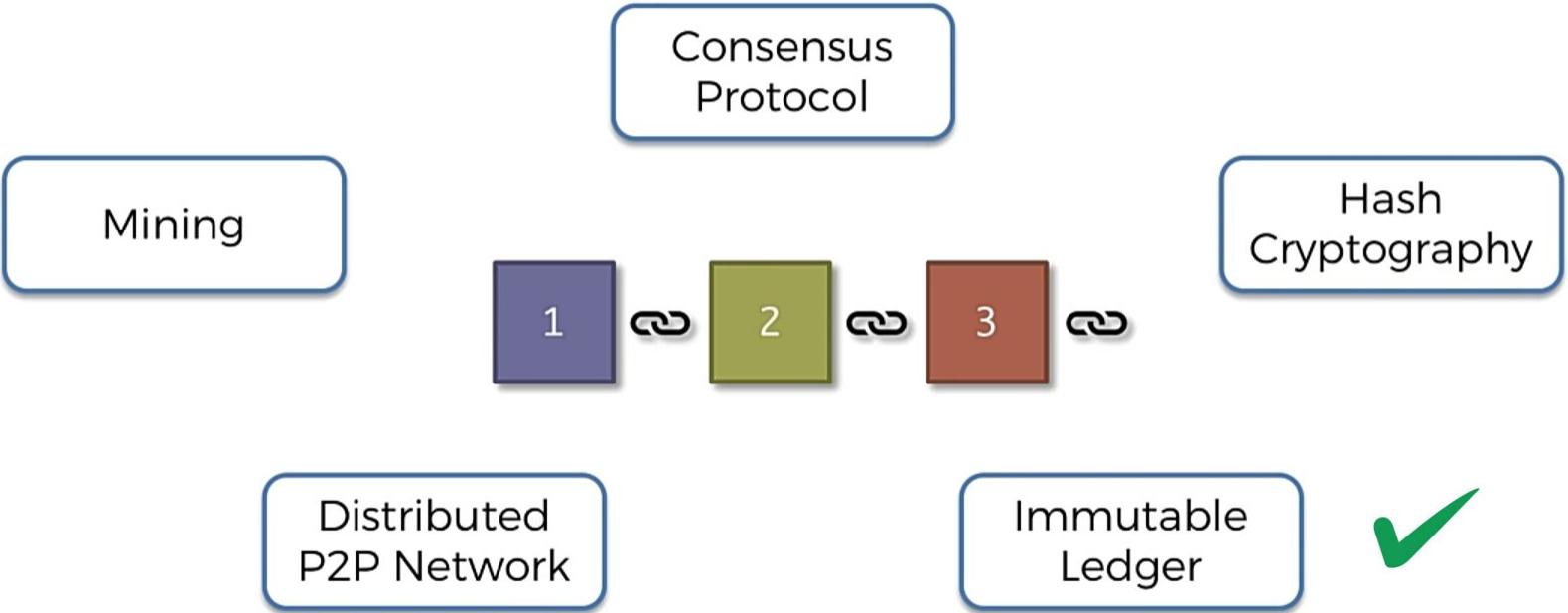
## 2. Blockchain Fundamentals

Let's take an example - If you use the SHA256 hash algorithm and pass 101 Blockchains as input, you will get the following output:

fbffd63a60374a31aa9811cbc80b577e23925a5874e86a17f712bab874f33ac9



## 2. Blockchain Fundamentals





## 2. Blockchain Fundamentals



Blockchain Demo: Public / Private Keys & Signing

Keys Signatures Transaction

### Transaction

Sign

Verify

Message

\$ 20.00

From:

048bcef76146dc920673d483b27e555e

->

04cc955bf8e359cc7ebbb66f4c2dc616

Signature

3044022038ce3cb35dd7d26956fae50585b300b40da4afd0575e4ee527dbd385b1f24d0c022055bd9624e35e6471954376f95201d90ec360d21917c

Verify



## 2. Blockchain Fundamentals



Blockchain Demo: Public / Private Keys & Signing

Keys Signatures Transaction

### Transaction

Sign Verify

Message

\$ 25.00

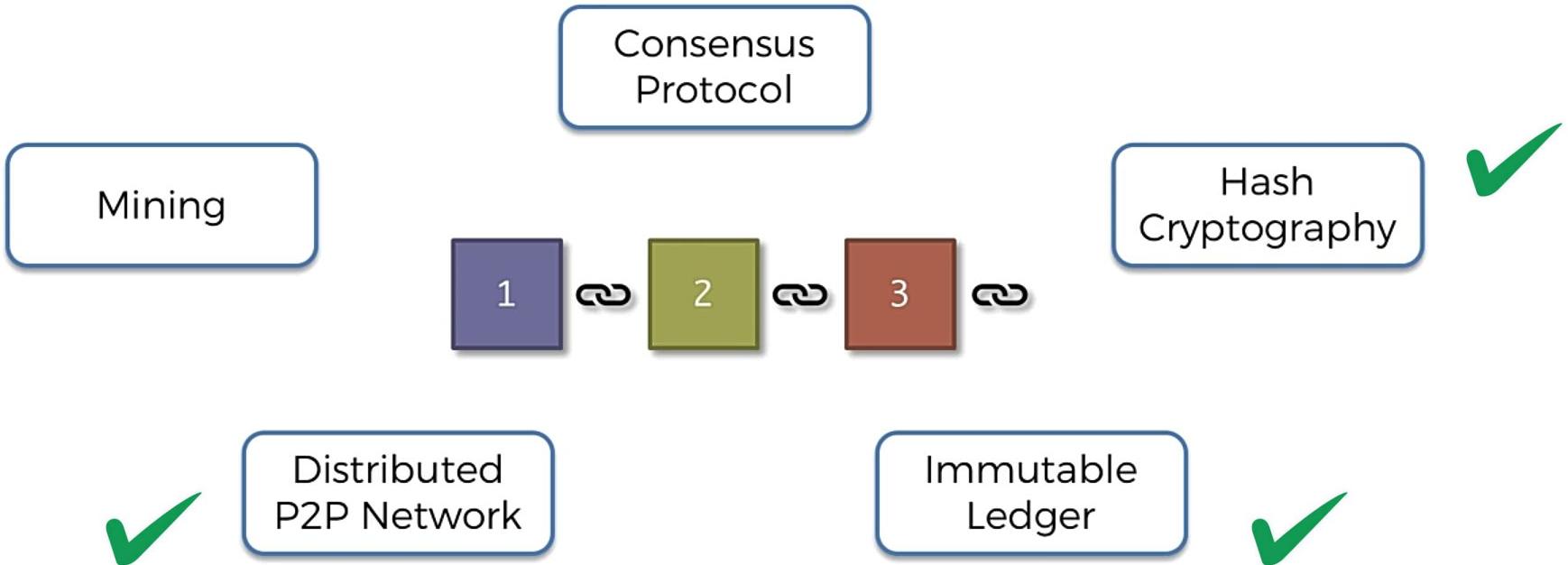
From: 048bcef76146dc920673d483b27e555e -> 04cc955bf8e359cc7ebbb66f4c2dc616.

Signature

3044022038ce3cb35dd7d26956fae50585b300b40da4af0575e4ee527dbd385b1f24d0c022055bd9624e35e6471954376f95201d90ec360d21917c

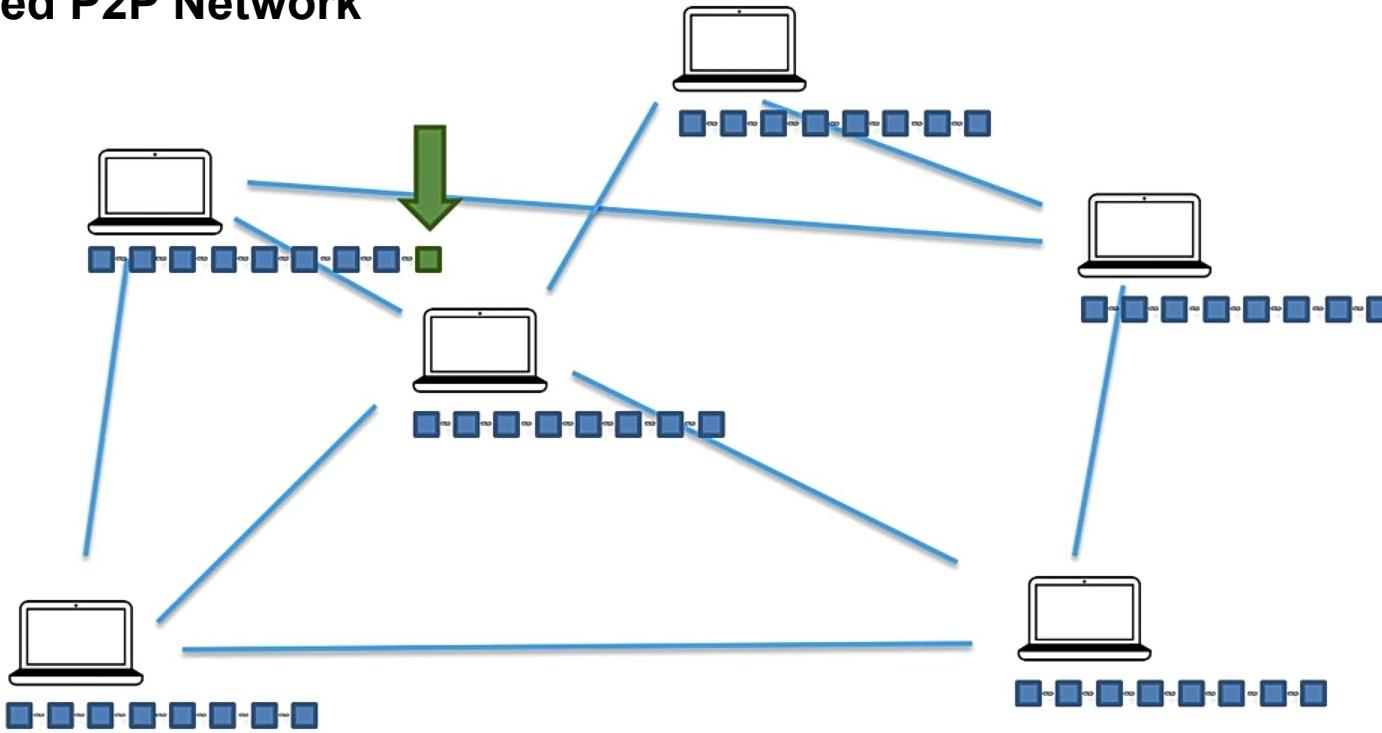
Verify

## 2. Blockchain Fundamentals



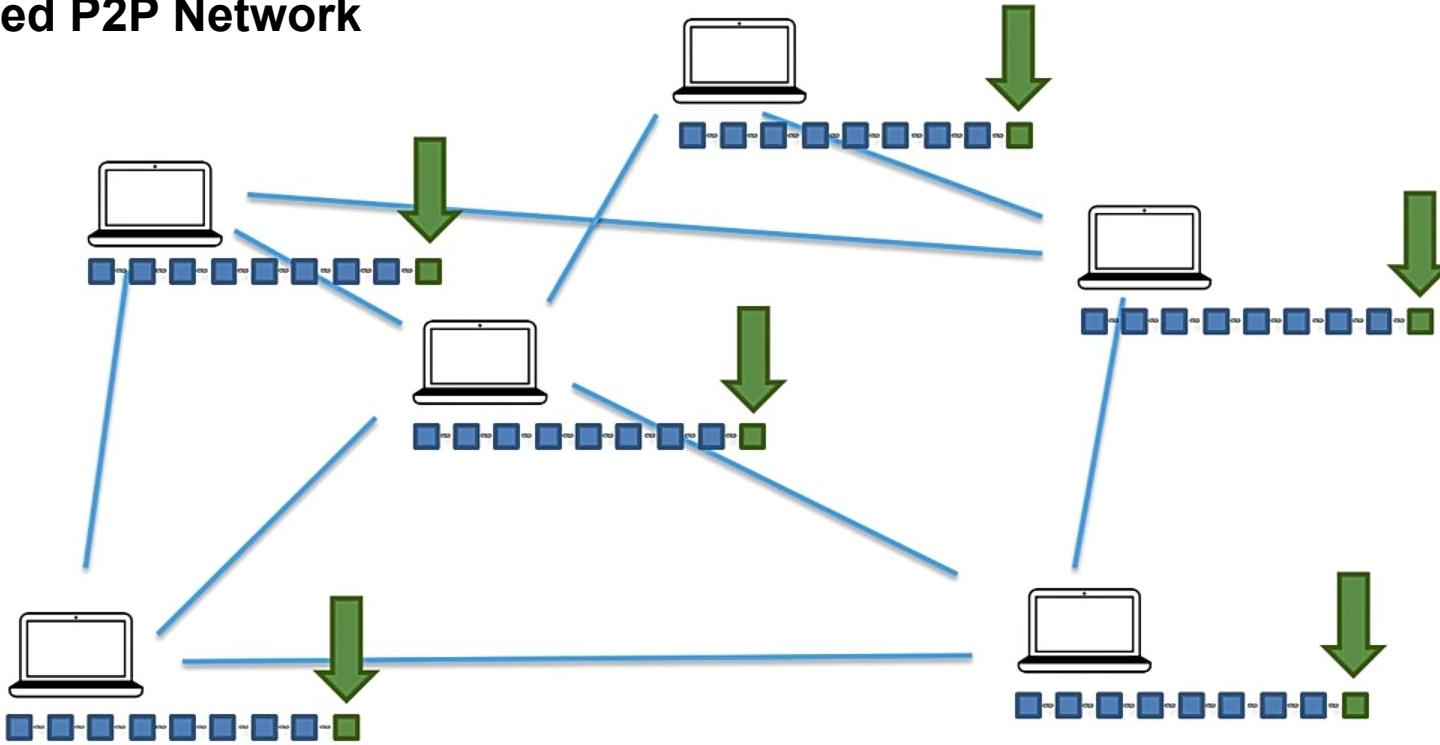
## 2. Blockchain Fundamentals

### Distributed P2P Network



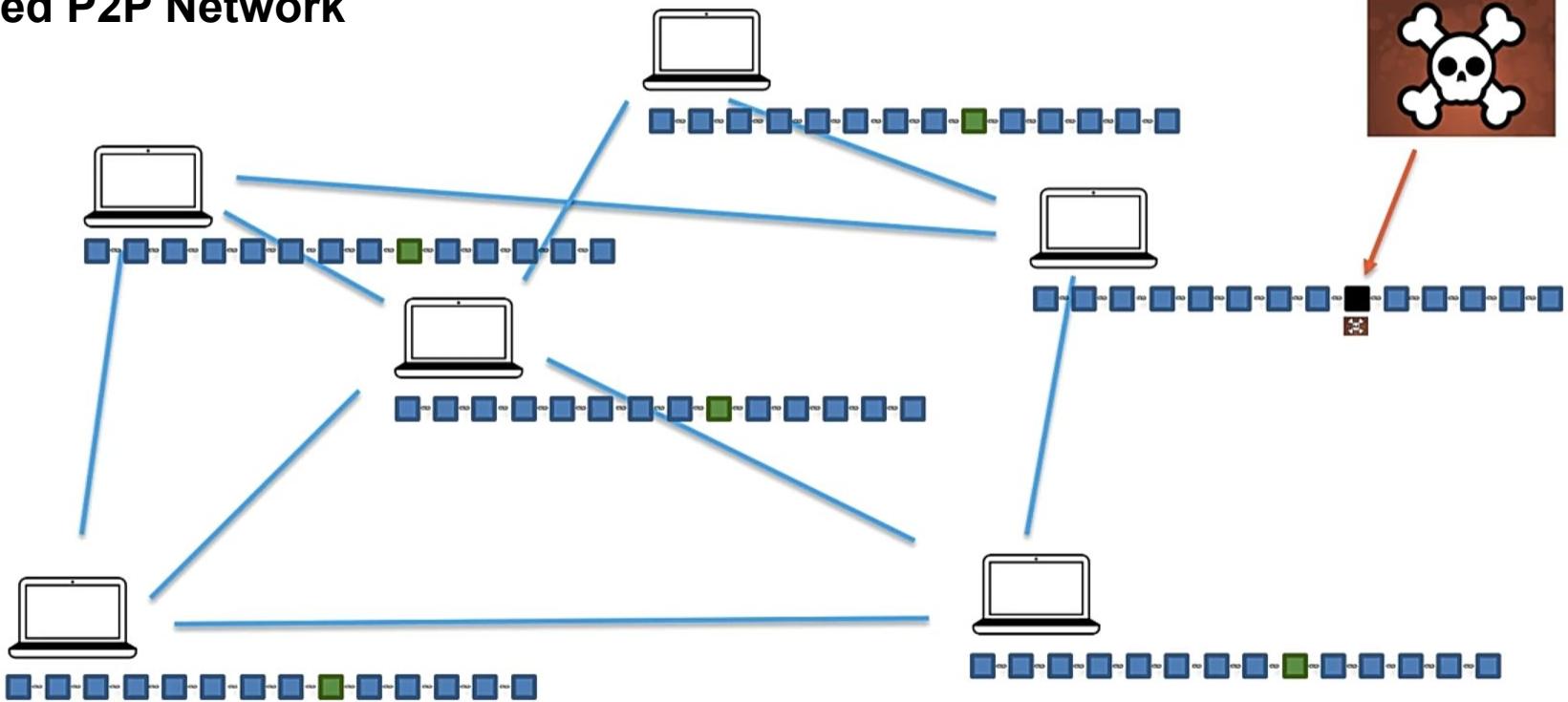
## 2. Blockchain Fundamentals

### Distributed P2P Network



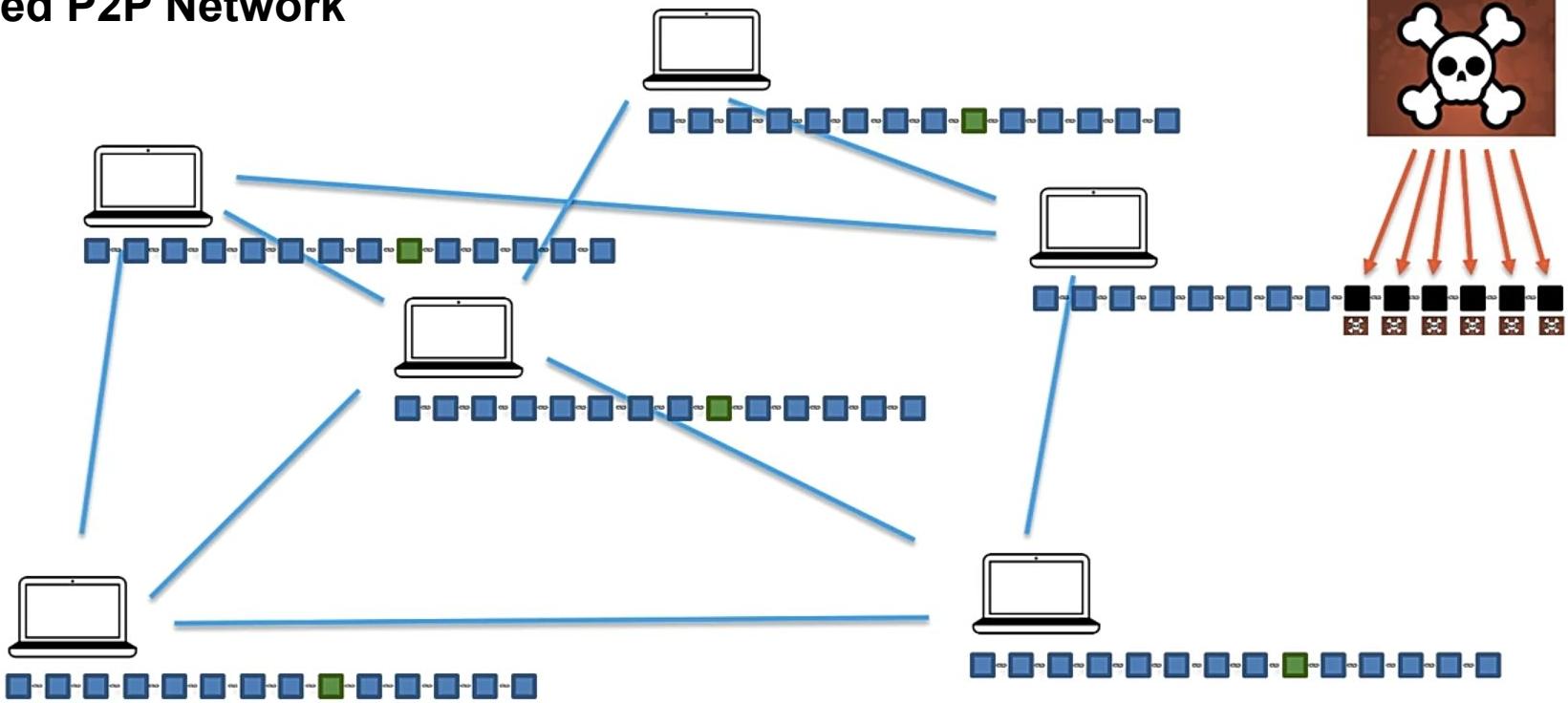
## 2. Blockchain Fundamentals

### Distributed P2P Network



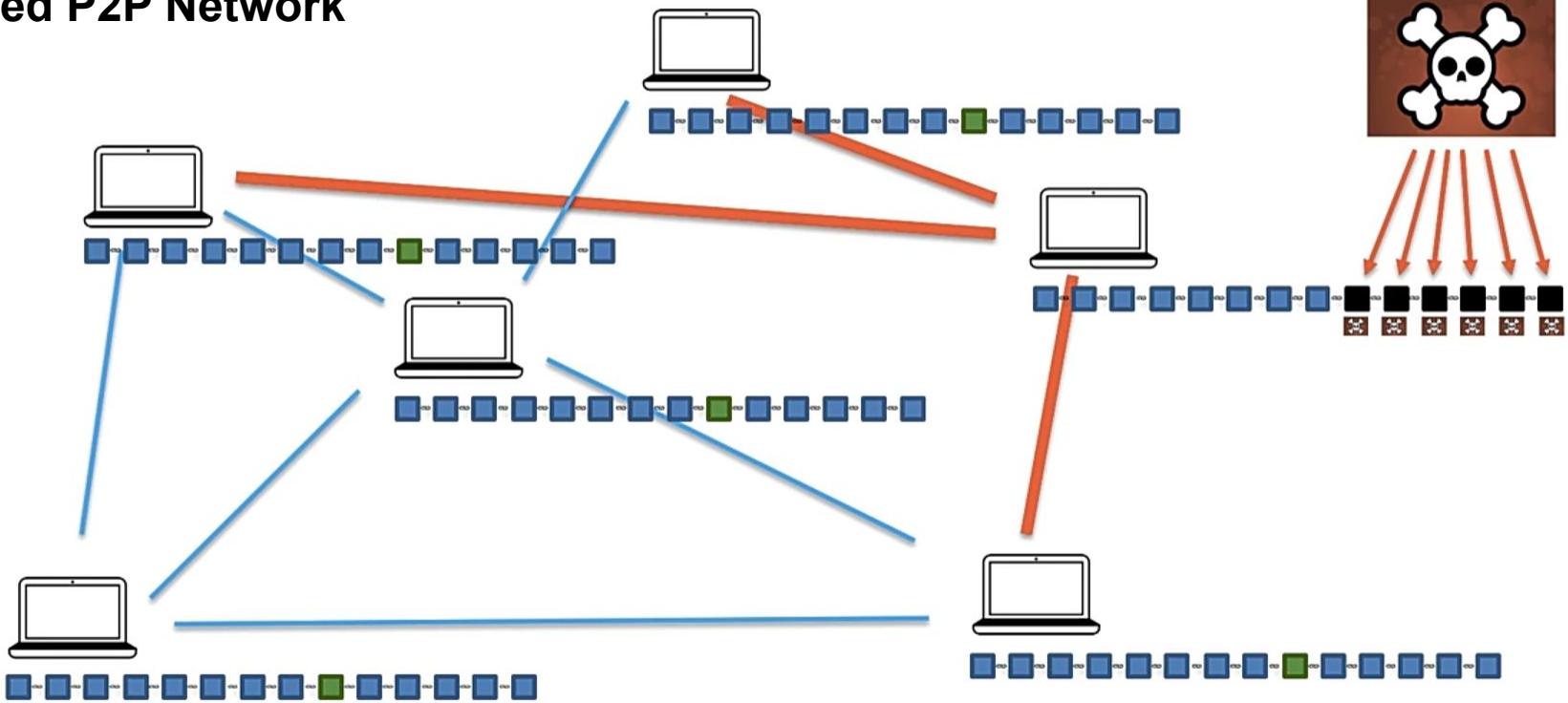
## 2. Blockchain Fundamentals

### Distributed P2P Network



## 2. Blockchain Fundamentals

### Distributed P2P Network



## 2. Blockchain Fundamentals

### Blockchain

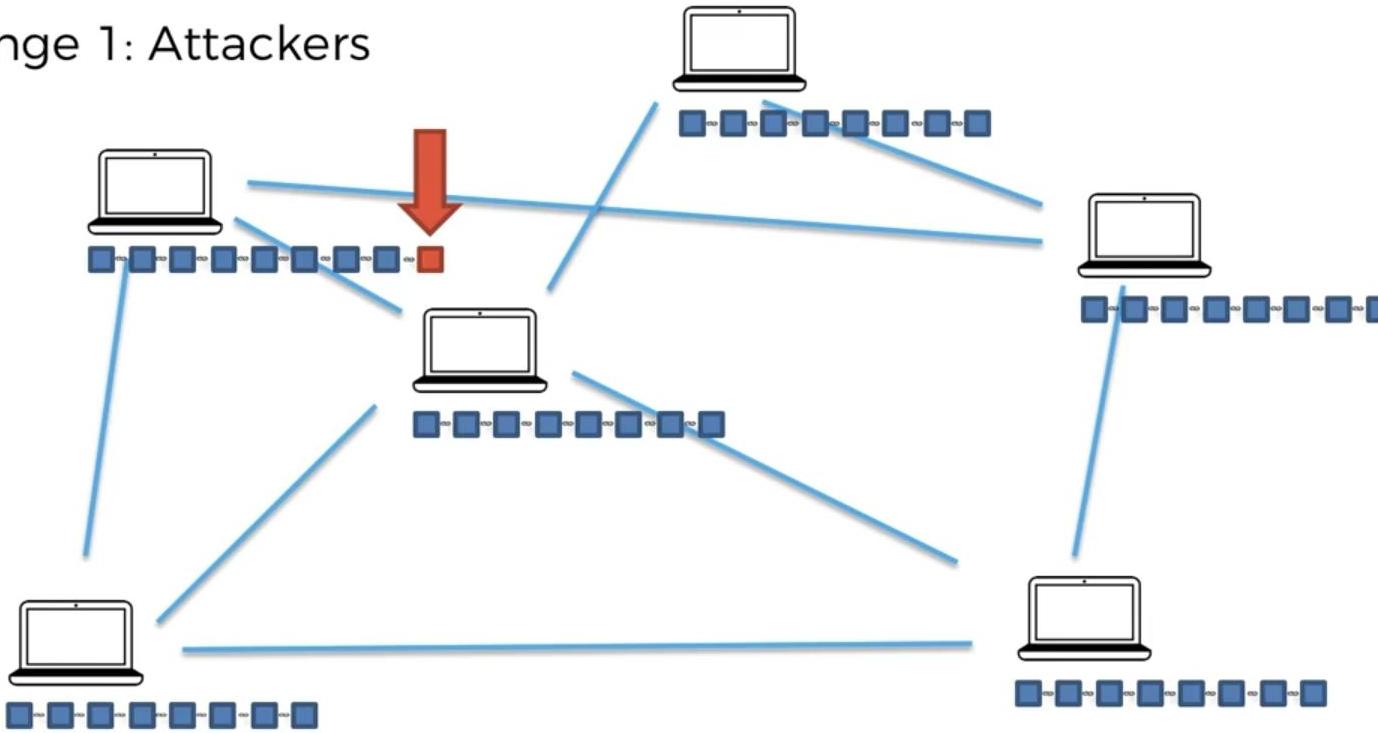
Block:	#	1	^
Nonce:	11316		
Data:			
Prev:	00		
Hash:	000015783b764259d382017d91a36d206d0600e2cbb3		
<button>Mine</button>			

Block:	#	2	^
Nonce:	35230		
Data:	Hai		
Prev:	000015783b764259d382017d91a36d206d0600e2cbb3		
Hash:	43301c373d54e3155c015cdad06a1e66d02df32ca7f4		
<button>Mine</button>			

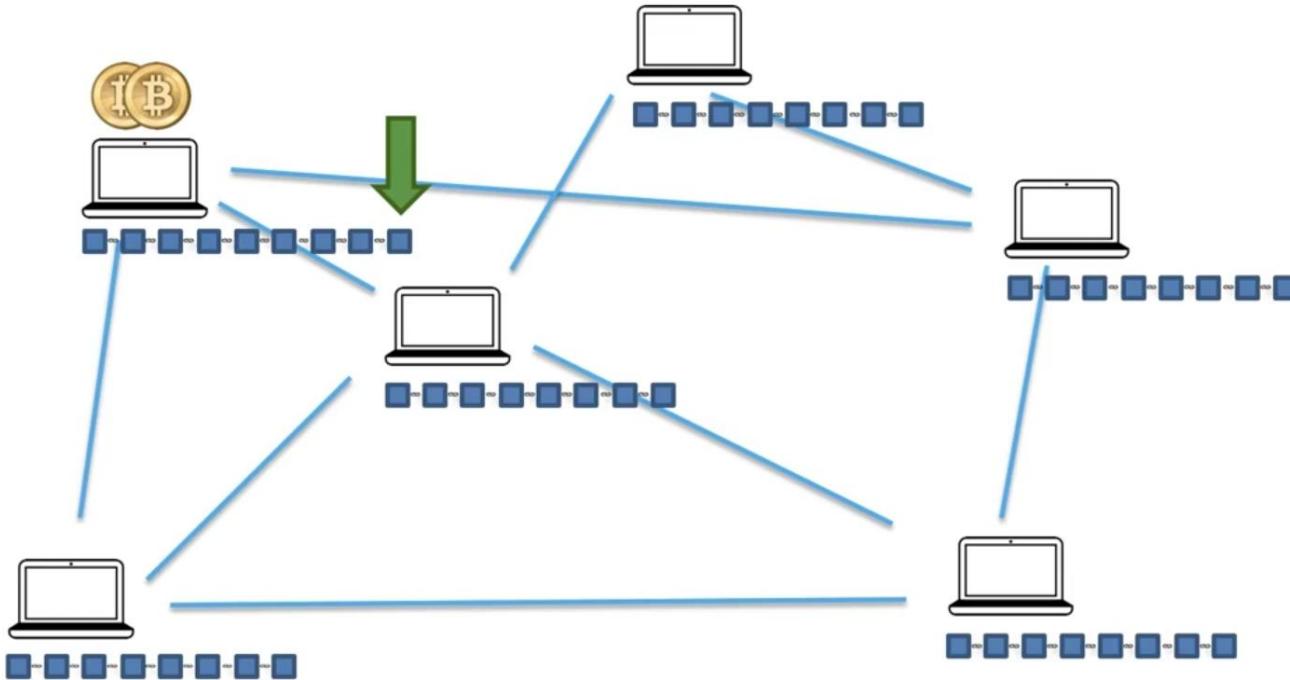
Block:	#	3	^
Nonce:	12937		
Data:			
Prev:	43301c373d54e3155c015cdad06a1e66d02df32ca7f4		
Hash:	37a9282dc5ad49d7387dc14838de		
<button>Mine</button>			

### Challenges in Distributed P2P Network

#### Challenge 1: Attackers



### Attackers Challenge in Distributed P2P Network addressed by Consensus

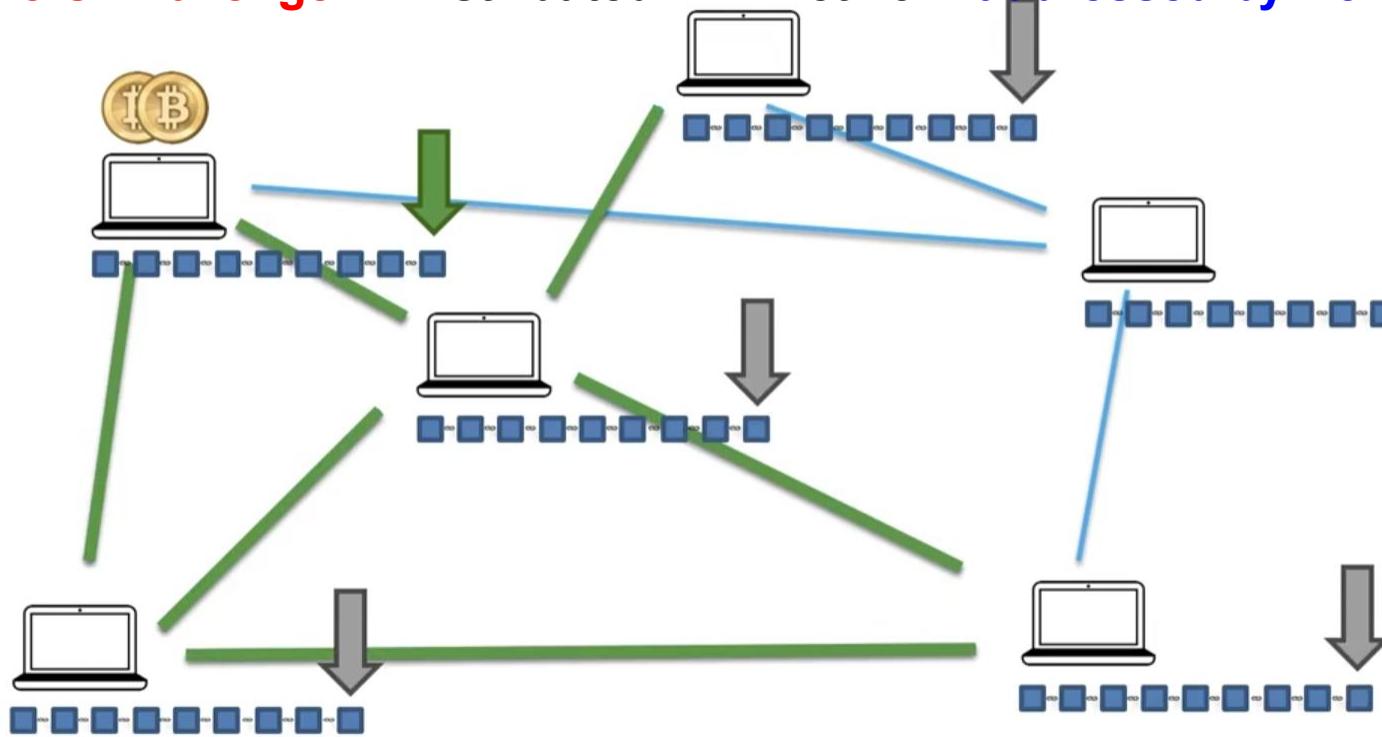


Miners get incentives for :

1. Adding a block
2. To play fair
3. From the transaction fees

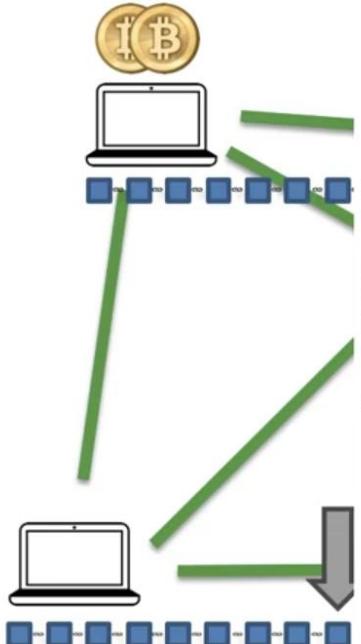
## 2. Blockchain Fundamentals

### Attackers Challenge in Distributed P2P Network addressed by Consensus



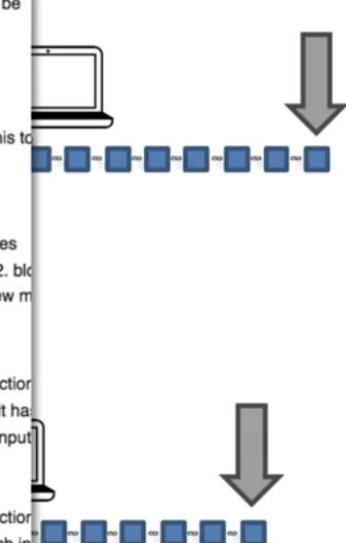
## 2. Blockchain Fundamentals

### Attackers Challenge in Distributed P2P Network addressed by Consensus



1. Check syntactic correctness
2. Reject if duplicate of block we have in any of the three categories
3. Transaction list must be non-empty
4. Block hash must satisfy claimed  $nBits$  proof of work
5. Block timestamp must not be more than two hours in the future
6. First transaction must be coinbase (i.e. only 1 input, with hash=0, n=-1), the rest must not be
7. For each transaction, apply "tx" checks 2-4
8. For the coinbase (first) transaction, scriptSig length must be 2-100
9. Reject if sum of transaction sig opcounts > MAX\_BLOCK\_SIGOPS
10. Verify Merkle hash
11. Check if prev block (matching prev hash) is in main branch or side branches. If not, add this to orphan block in prev chain; done with block
12. Check that  $nBits$  value matches the difficulty rules
13. Reject if timestamp is the median time of the last 11 blocks or before
14. For certain old blocks (i.e. on initial block download) check that hash matches known values
15. Add block into the tree. There are three cases: 1. block further extends the main branch; 2. block make it become the new main branch; 3. block extends a side branch and makes it the new main branch
16. For case 1, adding to main branch:
  1. For all but the coinbase transaction, apply the following:
    1. For each input, look in the main branch to find the referenced output transaction
    2. For each input, if we are using the  $r$ th output of the earlier transaction, but it has been spent, reject
    3. For each input, if the referenced output transaction is coinbase (i.e. only 1 input) and has less than 100 confirmations; else reject.
    4. Verify crypto signatures for each input; reject if any are bad
    5. For each input, if the referenced output has already been spent by a transaction, reject
    6. Using the referenced output transactions to get input values, check that each input value is less than or equal to the sum of output values
    7. Reject if the sum of input values < sum of output values
  2. Reject if coinbase value > sum of block creation fee and transaction fees

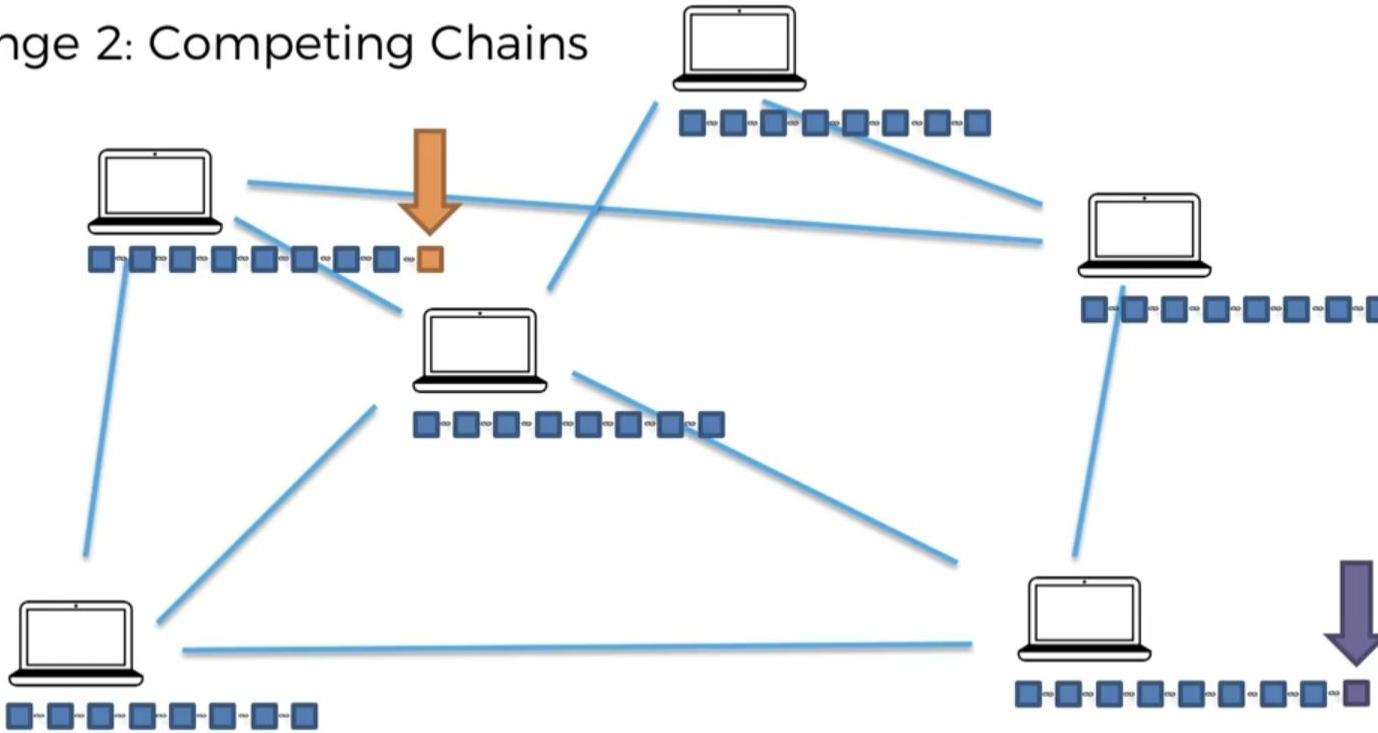
Cryptographic puzzles:  
Hard to solve - Easy to verify



## 2. Blockchain Fundamentals

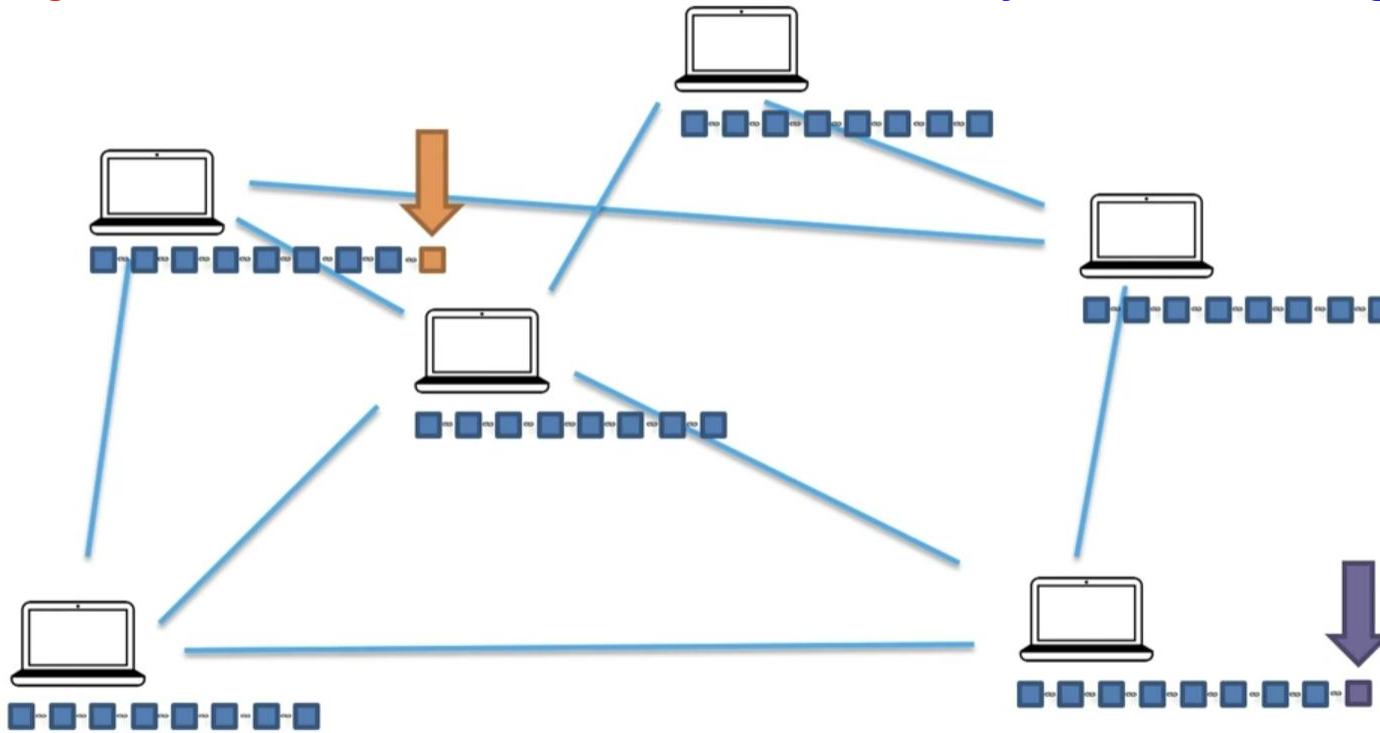
### Challenges in Distributed P2P Network

#### Challenge 2: Competing Chains



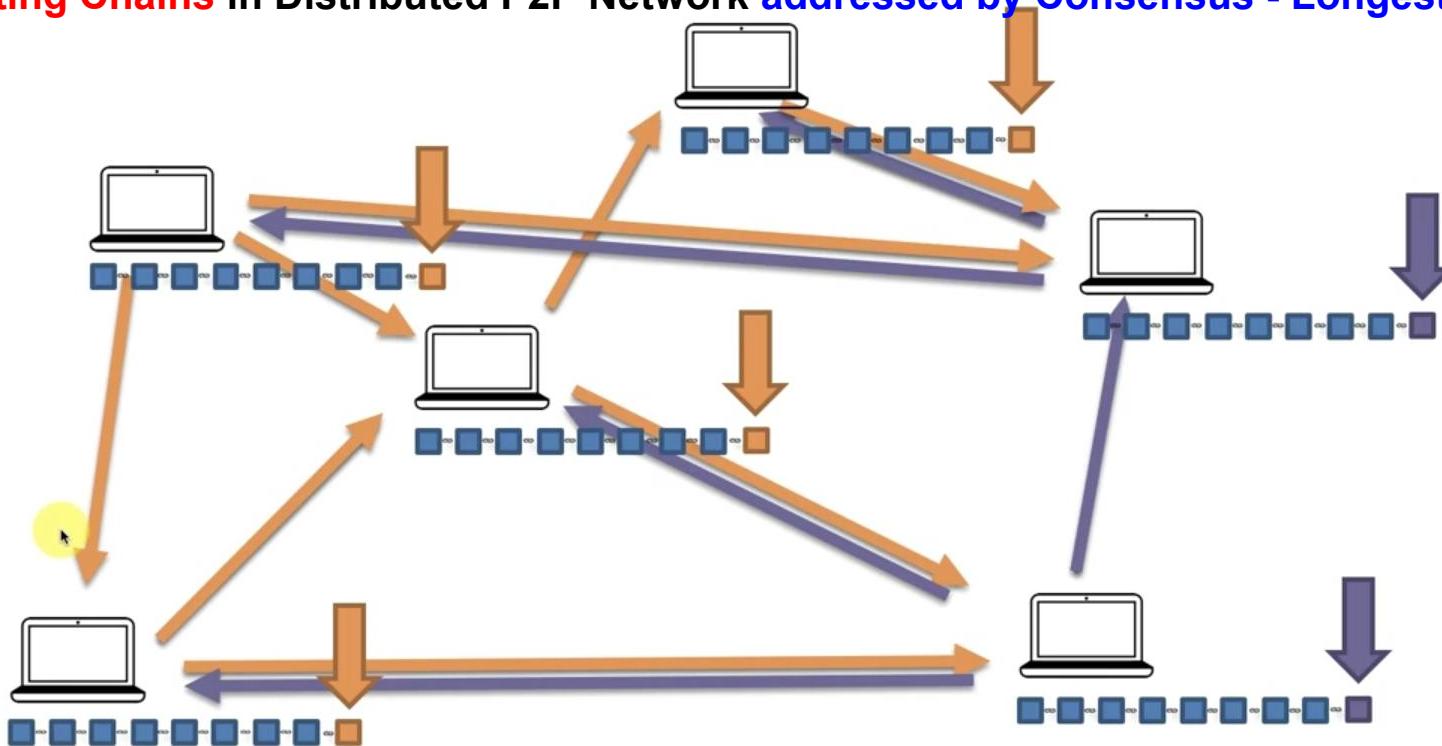
## 2. Blockchain Fundamentals

Competing Chains in Distributed P2P Network addressed by Consensus - Longest Chain



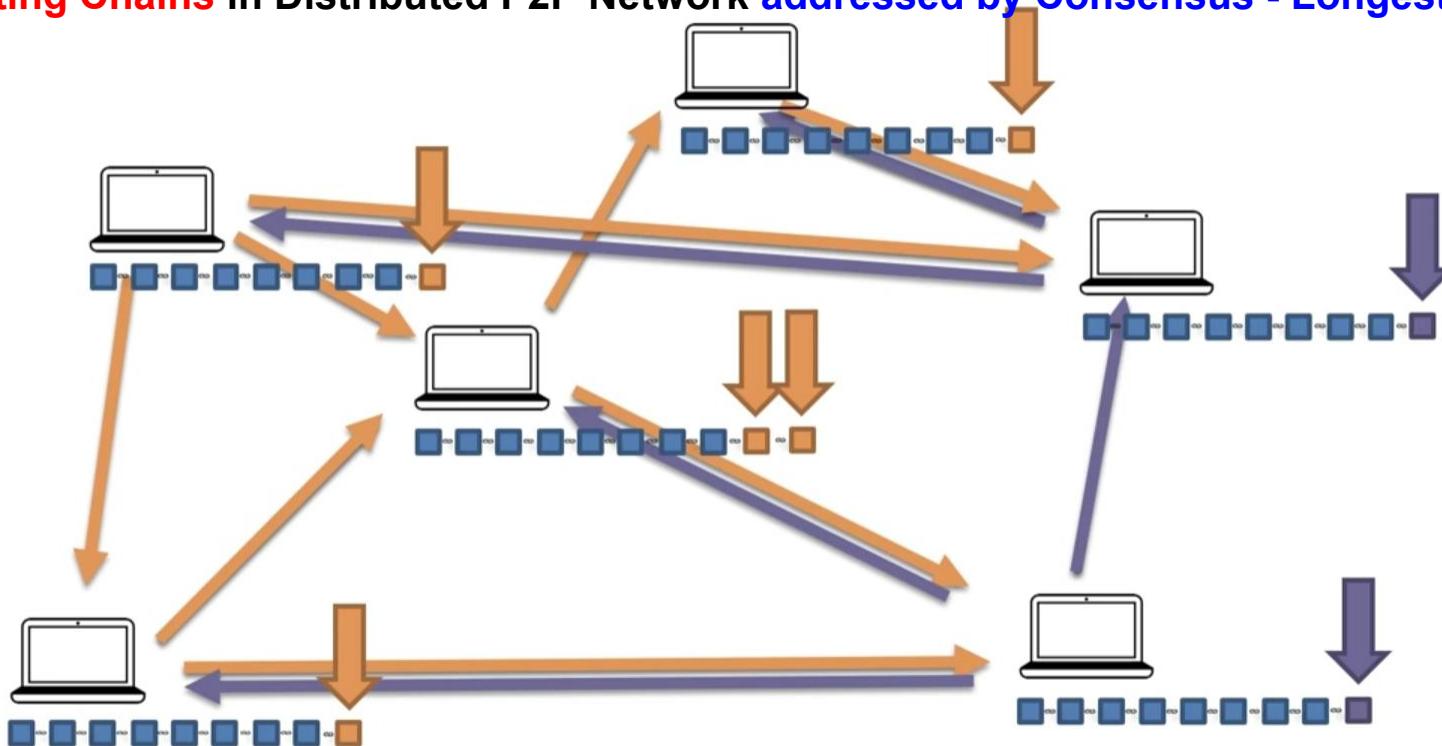
## 2. Blockchain Fundamentals

Competing Chains in Distributed P2P Network addressed by Consensus - Longest Chain



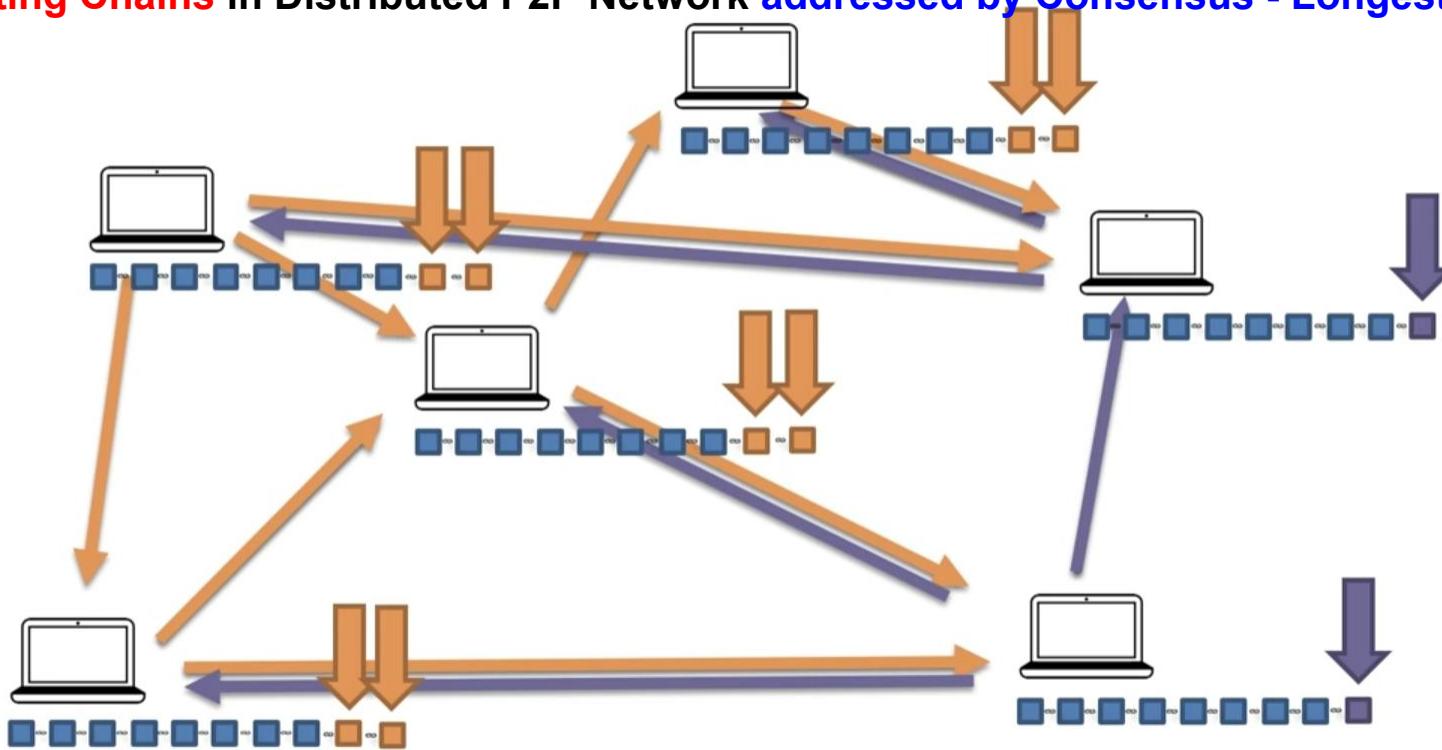
## 2. Blockchain Fundamentals

Competing Chains in Distributed P2P Network addressed by Consensus - Longest Chain



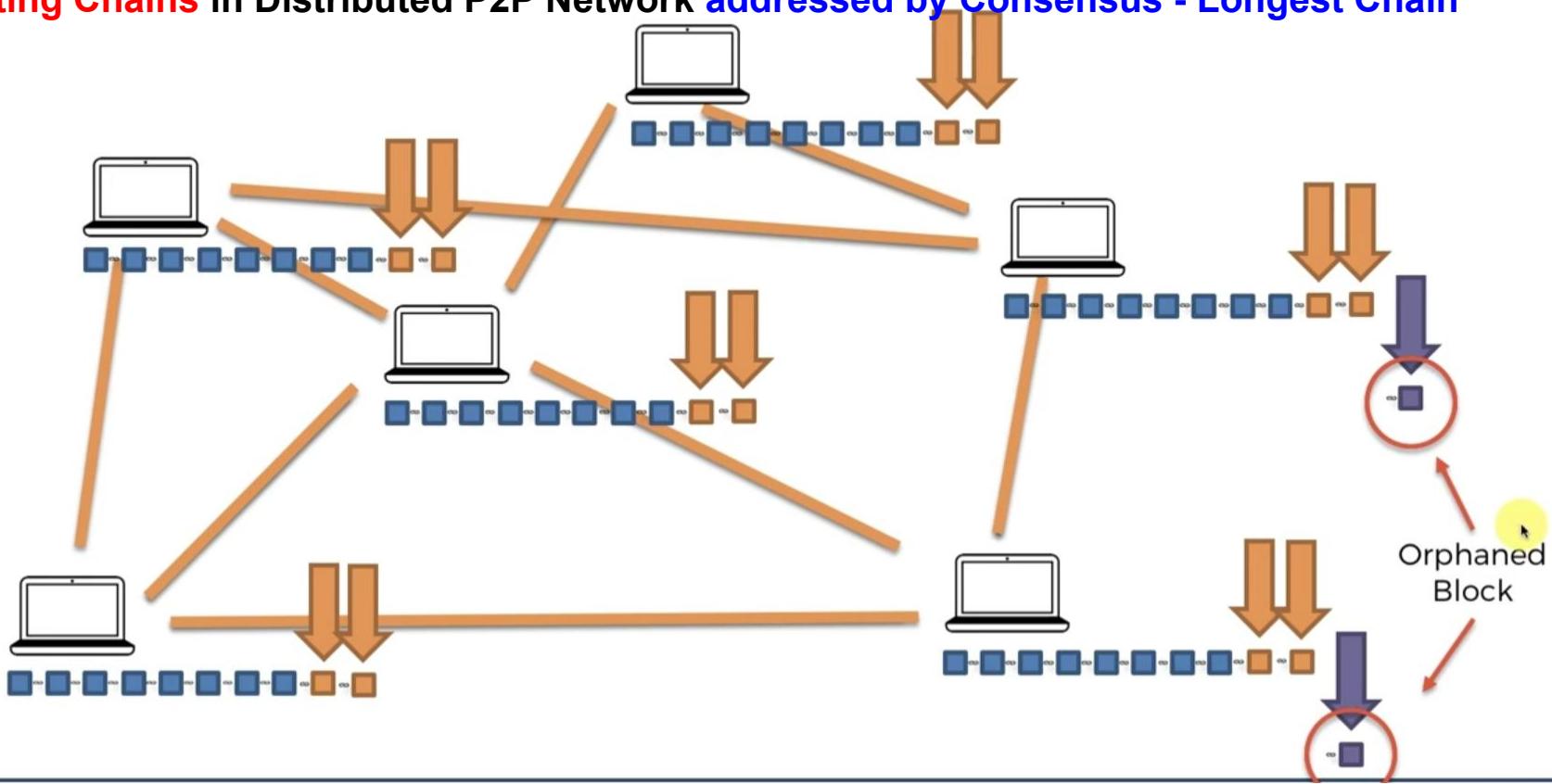
## 2. Blockchain Fundamentals

Competing Chains in Distributed P2P Network addressed by Consensus - Longest Chain

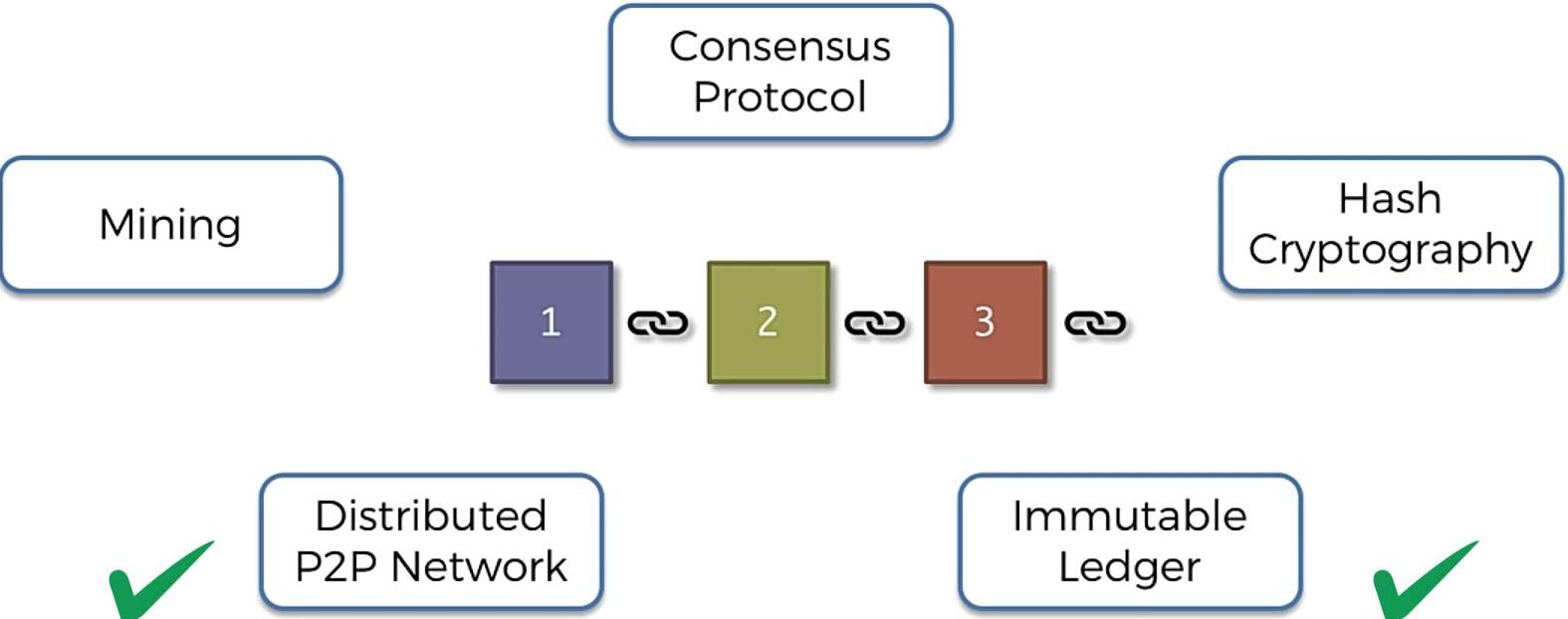


## 2. Blockchain Fundamentals

Competing Chains in Distributed P2P Network addressed by Consensus - Longest Chain



## 2. Blockchain Fundamentals



## 2. Blockchain Fundamentals

### How Mining Works ?

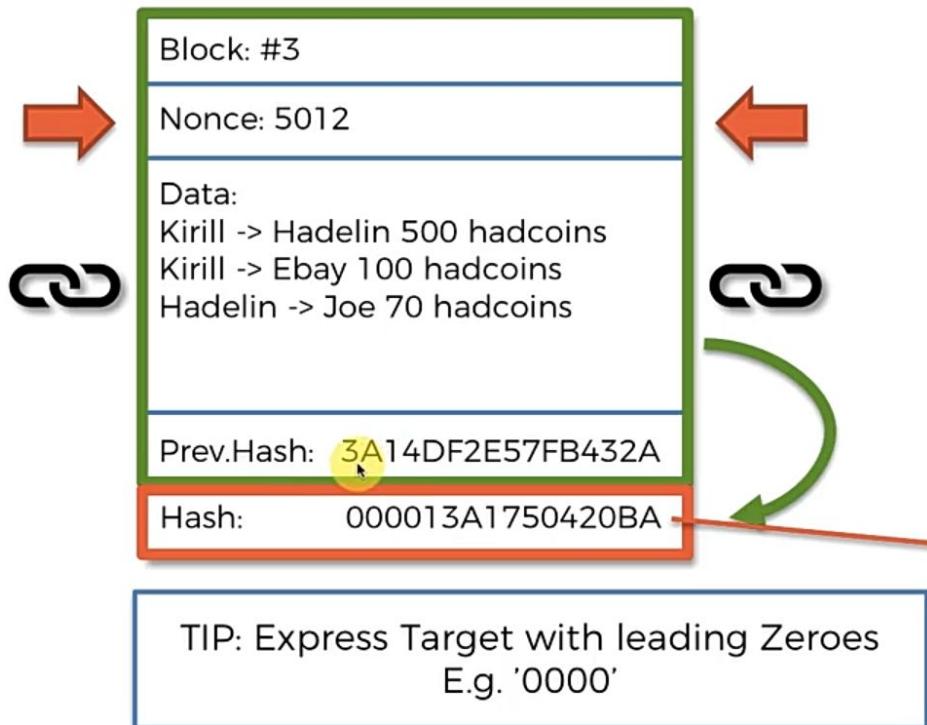


### How Mining Works ?

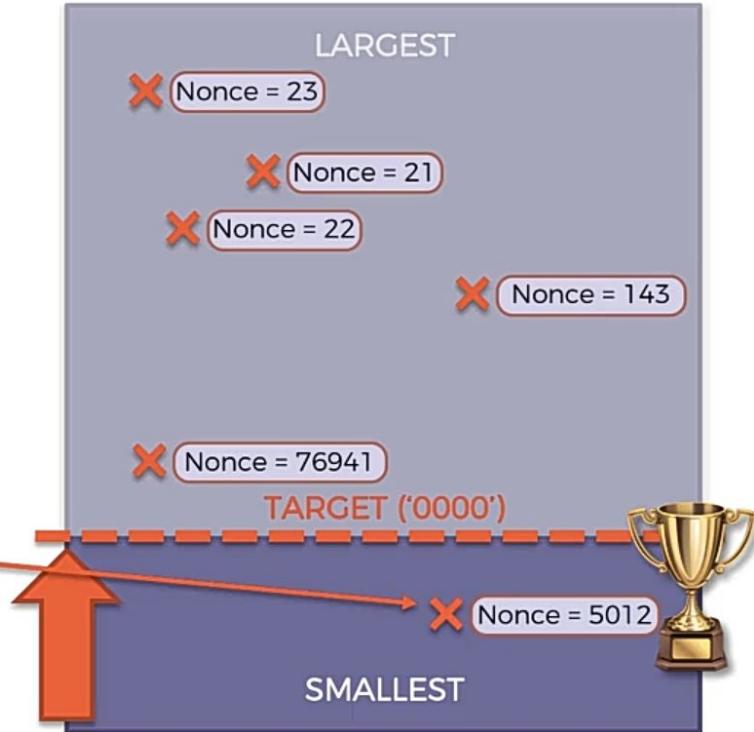


## 2. Blockchain Fundamentals

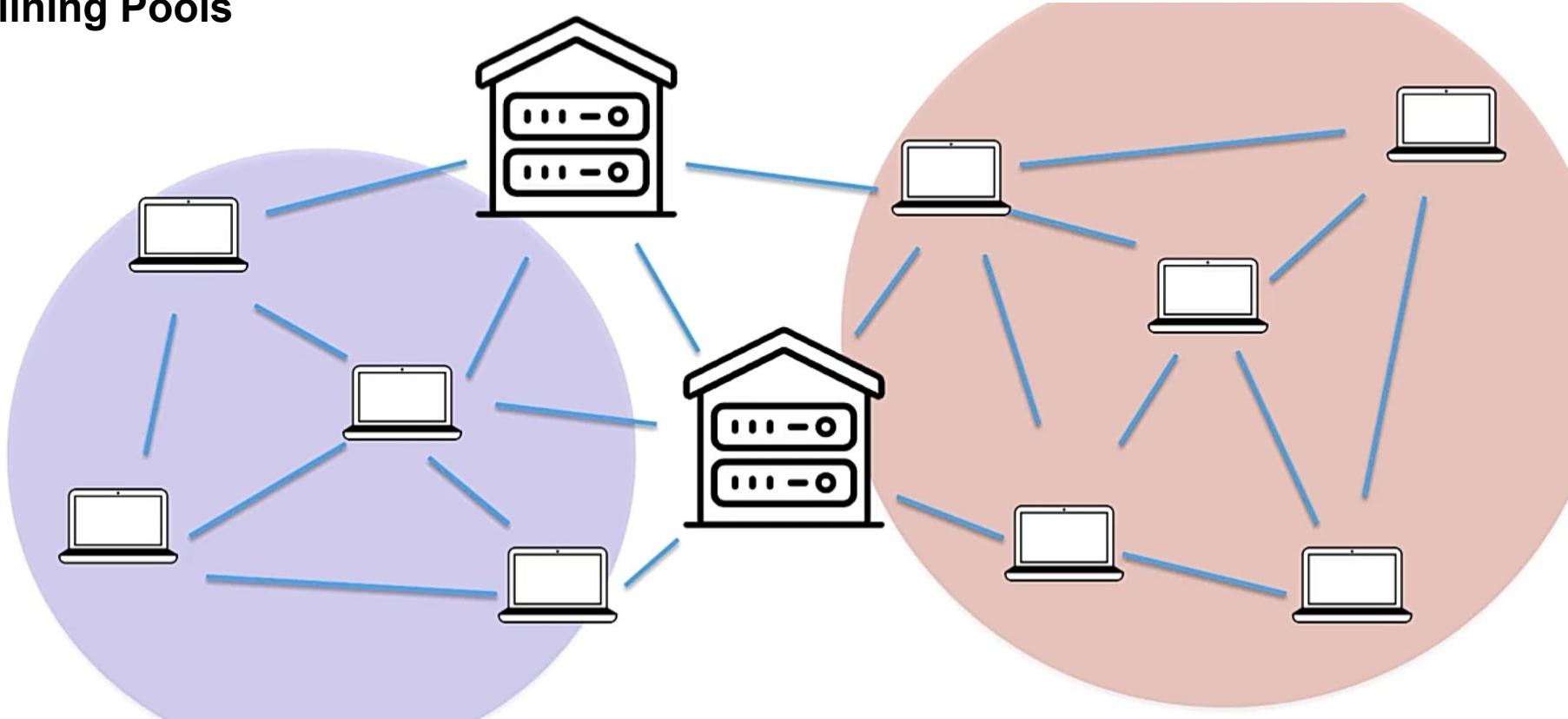
### How Mining Works ?



### - ALL POSSIBLE HASHES -



### Mining Pools



## 2. Blockchain Fundamentals

### Mining Pools

Hi! Sign in or register | Daily Deals | Gift Cards | Help & Contact 

Sell | My eBay  

**ebay** Shop by category ▾ Search for anything All Categories ▾ **Search** Advanced

eBay > Coins & Paper Money > Virtual Currency > Miners Share

### Cryptocurrency GPU Mining Rig 3x GTX 1080 TI Ethereum Zcash Bitcoin Extras

★★★★★ 2 product ratings | About this product



9 viewed per hour

New (other): lowest price

**\$5,599.00**  
+ \$549.95 Shipping

Get it by Mon, Mar 5 - Thu, Apr 12 from New Baltimore, Michigan

- New other (see details) condition
- No returns, but backed by eBay Money back guarantee

"New  
Easily Mine Zcash or Other Equihash Coins at 2250 Sol/s (2250 h/s) @ 890W. Mine Zcash (ZEC), Bitcoin Gold (BTG),..."  
[Read full description](#)

[See details >](#)

Qty : 1

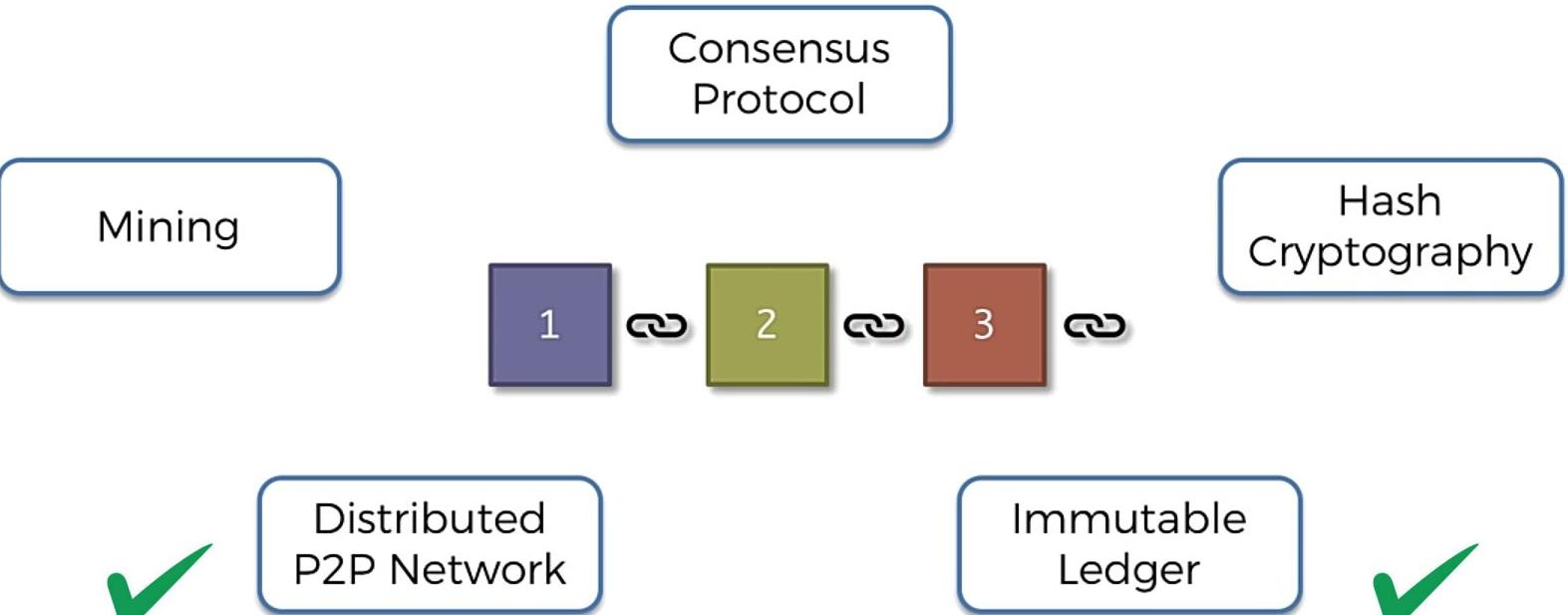
**Buy It Now**

**Add to cart**

**Watch**

Sold by [partdiscounter \(42407\)](#)  
99.8% Positive feedback

## 2. Blockchain Fundamentals



### What is Consensus?

- As per Webster dictionary, a consensus is a **general agreement or opinion shared by all the people in a group.**
- A protocol is a **system of standard rules that are acceptable by all parties** to control the exchange of information in a network. Thus, a **consensus protocol** in Blockchain can be defined as **a set of rules and procedures for attaining a unified agreement (consensus) between the participating nodes** on the status of the network.
- The consensus protocol **aims to overcome the classic problem of a distributed computing system known as the Byzantine Generals Problem**

## 2. Blockchain Fundamentals

### Different Consensus Mechanisms



Proof of History  
(PoH)



Proof of  
Importance  
(PoI)



Proof of Work  
(PoW)



Proof of Stake  
(PoS)

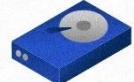


Proof of  
Elapsed Time  
(PoET)

### DIFFERENT TYPES OF CONSENSUS MECHANISMS



Delegated  
Proof of Stake  
(DPoS)



Proof of Capacity/  
Proof of Space  
(PoC/PoSpace)



Proof of Burn  
(PoB)



Proof of Authority  
(PoA)



Proof of Activity  
(PoA)

## 2. Blockchain Fundamentals

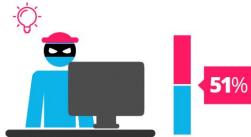
### Proof of Work      vs      Proof of Stake



*Proof of work is a requirement to define an expensive computer calculation, also called mining*



*Proof of stake, the creator of a new block is chosen in a deterministic way, depending on its wealth, also defined as stake.*



*A reward is given to the first miner who solves each blocks problem.*



*The PoS system there is no block reward, so, the miners take the transaction fees.*



*Network miners compete to be the first to find a solution for the mathematical problem*



*Proof of Stake currencies can be several thousand times more cost effective.*

## 2. Blockchain Fundamentals

	Public	Private	Hybrid	Consortium
 <b>Permissioned/ Permissionless</b>	Permissionless	Permissioned	Permissioned & Permissionless	Permissioned
 <b>Control</b>	No control by a central authority	Control by a central authority	Control by a central authority	Control by multiple central authorities
 <b>Main Advantages</b>	✓ Independence ✓ Transparency	✓ Performance ✓ Scalability	✓ Performance ✓ Low Cost	✓ Performance ✓ Security
 <b>Main Disadvantages</b>	✗ Performance ✗ Scalability Issues	✗ Security ✗ Trust	✗ Transparency ✗ Upgrading	✗ Transparency
 <b>Examples</b>	Bitcoin Litecoin	Hyperledger Fabric	XRP token	Corda Quorum

## 2. Blockchain Fundamentals



Feature	Bitcoin	Ethereum
Purpose	Digital currency	Smart contract platform
Launch	2009	2015
Founder	Satoshi Nakamoto	Vitalik Buterin
Consensus	Proof of Work	Proof of Stake
Supply	Fixed (21M)	No fixed cap



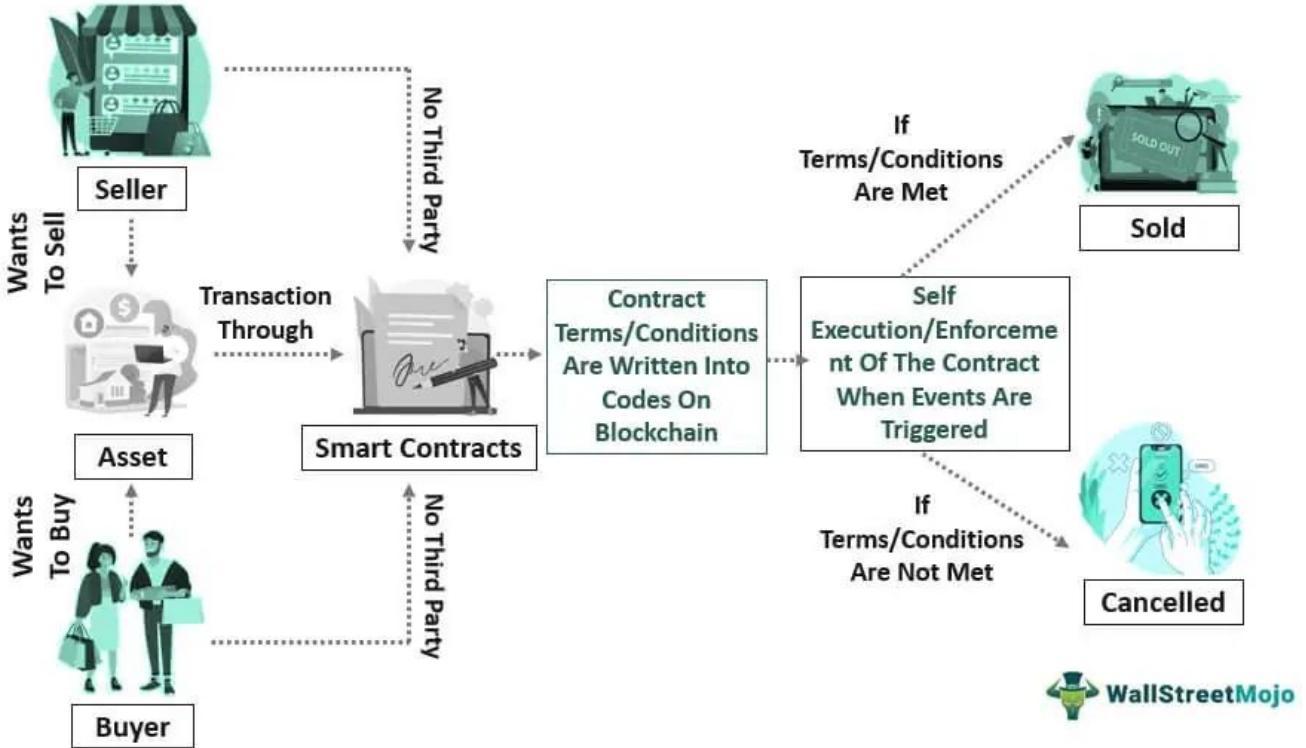
## 2. Blockchain Fundamentals



### What is a Smart Contract ?

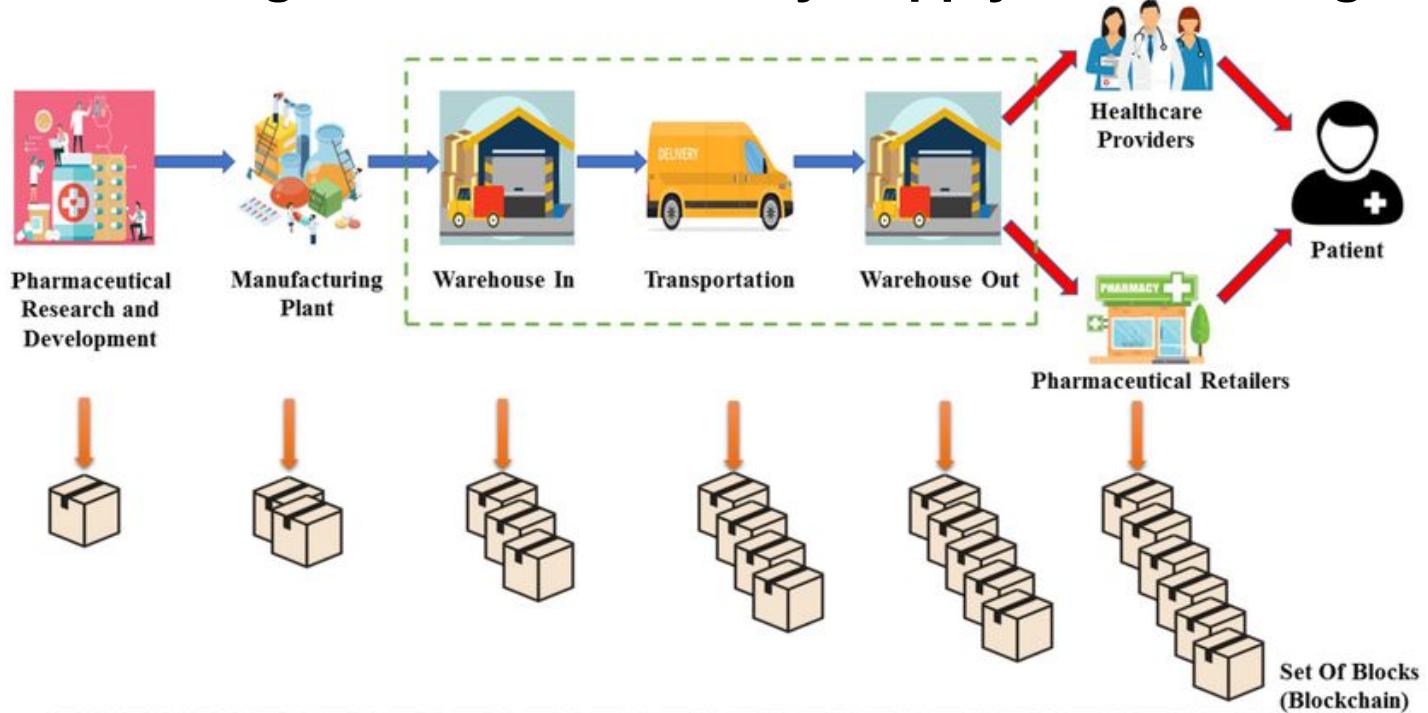
- A **self-executing program** that automates the actions required in an agreement or contract. Once completed, the transactions are trackable and irreversible.
- Smart contracts permit **trusted transactions and agreements to be carried out among disparate, anonymous parties** without the need for a central authority, legal system, or external enforcement mechanism.
- **do not contain legal language**, terms, or agreements
- **Only code that executes actions when specified conditions** are met.
- Nick Szabo, an American computer scientist
  - invented a virtual currency called "**Bit Gold**" in 1998,
  - defined smart contracts as computerized transaction protocols that execute the terms of a contract

### Smart Contract Functioning



## 2. Blockchain Fundamentals

### Blockchain Integration into Pharmacy Supply Chain Management



# Topics to be covered

1. Introduction to Blockchain
2. Blockchain Fundamentals
- 3. Cryptocurrency Basics**
4. Introduction to Solidity Programming



### 3. Cryptocurrency Basics

#### Cryptocurrency as a **Form of Currency**

- **Cryptos** - Eg. **Bitcoin, Litecoin, Shiba Inu, Dogecoin etc...**
- Various companies and even countries around the world accept some of these digital currencies for conducting transactions.
- However, the **high volatility** of Bitcoin and other popular cryptocurrencies makes it unsuitable for everyday use by the public.

#### Benefits:

- **Faster global transfers** with reduced settlement times
- **Lower transaction fees** eliminating intermediaries
- **Enhanced security** through decentralization and cryptography
- **Financial inclusion** for unbanked populations
- **Programmability** enabling smart contracts and automation

### 3. Cryptocurrency Basics

#### List of Latest Cryptocurrencies to Invest in 2025!

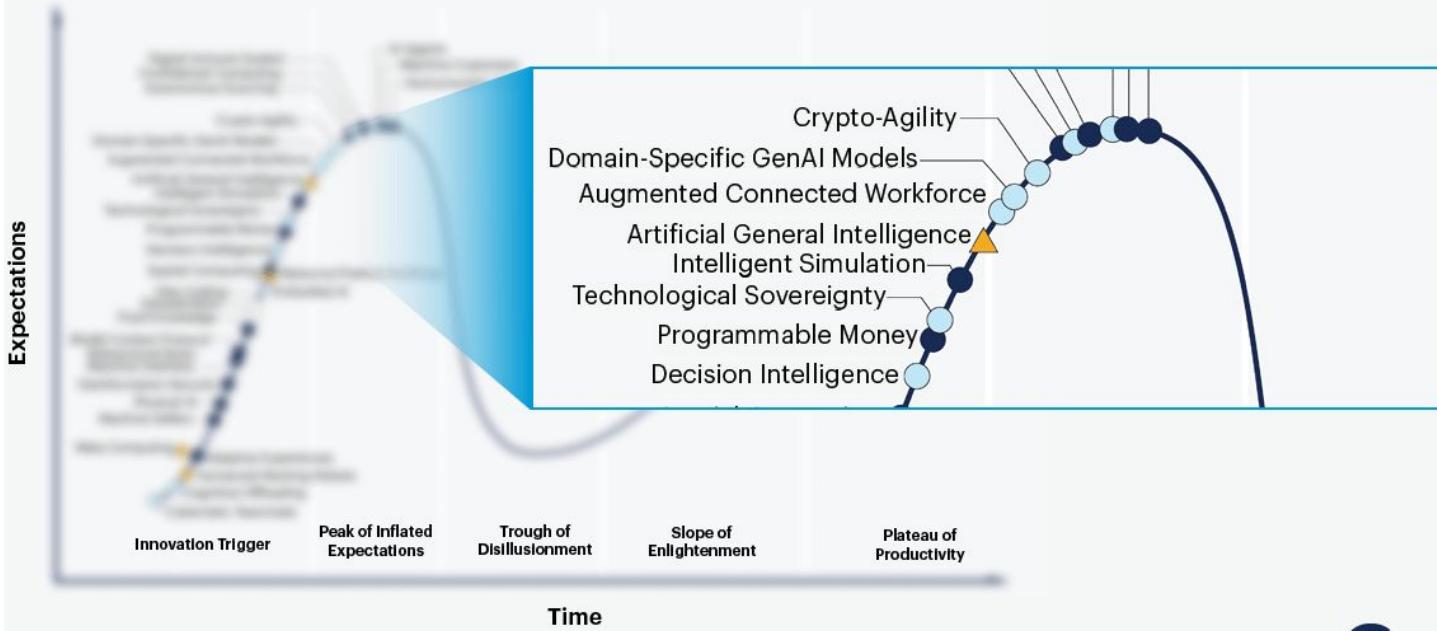
RANK	APPS	RANK	APPS		
1		Bitcoin	11		Litecoin
2		Ethereum	12		Chainlink
3		Ripple	13		Tron
4		Solana	14		Tether
5		Dogecoin	15		Dai
6		Binance Coin	16		SKALE
7		Cardano	17		PEPE
8		Avalanche	18		Uniswap
9		Polkadot	19		Bitcoin Cash
10		Stellar	20		Cosmos

### 3. Cryptocurrency Basics

## Hype Cycle of Emerging Technologies, 2025

Plateau will be reached:

- < 2 years
- 2 – 5 years
- 5 – 10 years
- ▲ >10 years
- ✗ obsolete before plateau



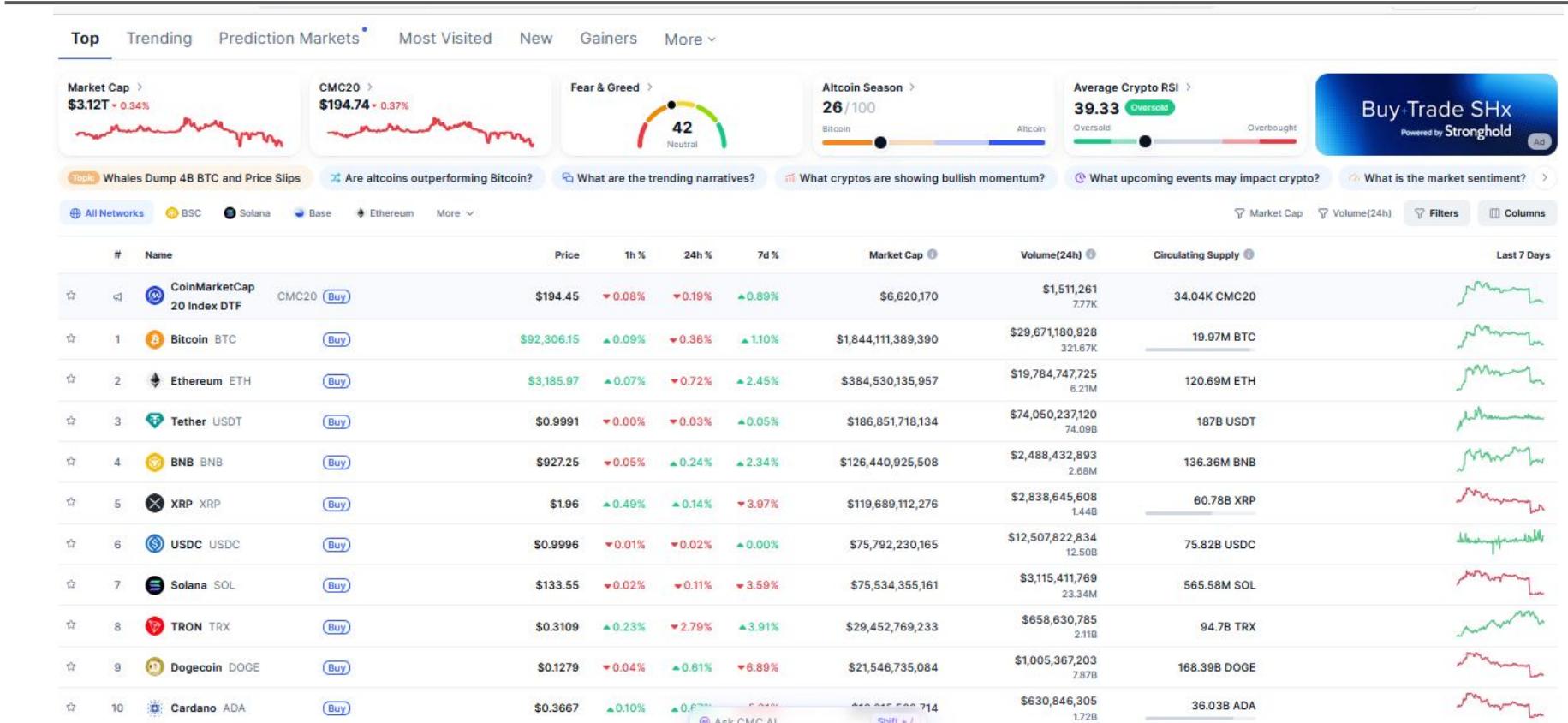
Source: Gartner

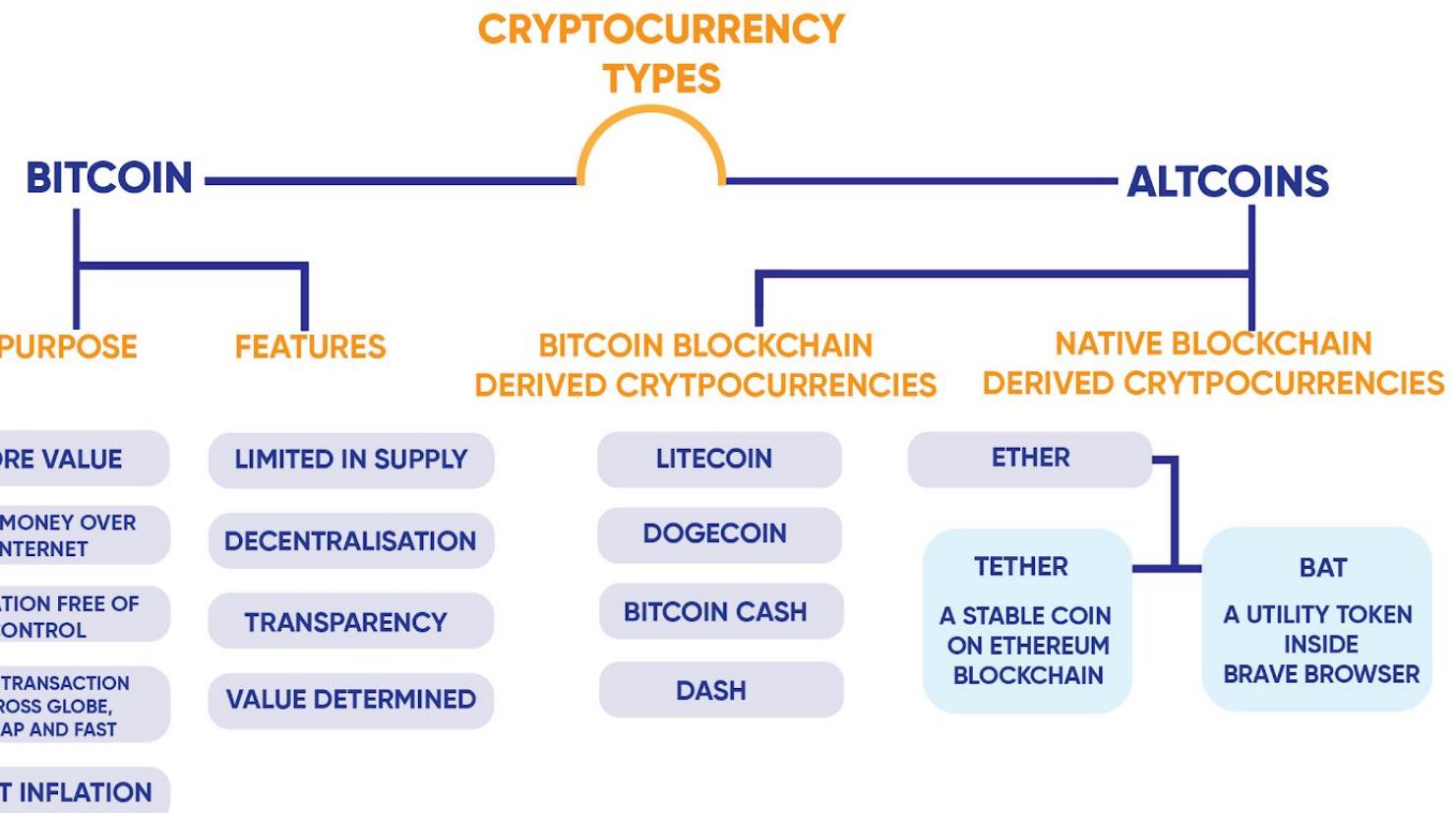
© Gartner, Inc. and/or its affiliates. All rights reserved. CTMKT\_3957950



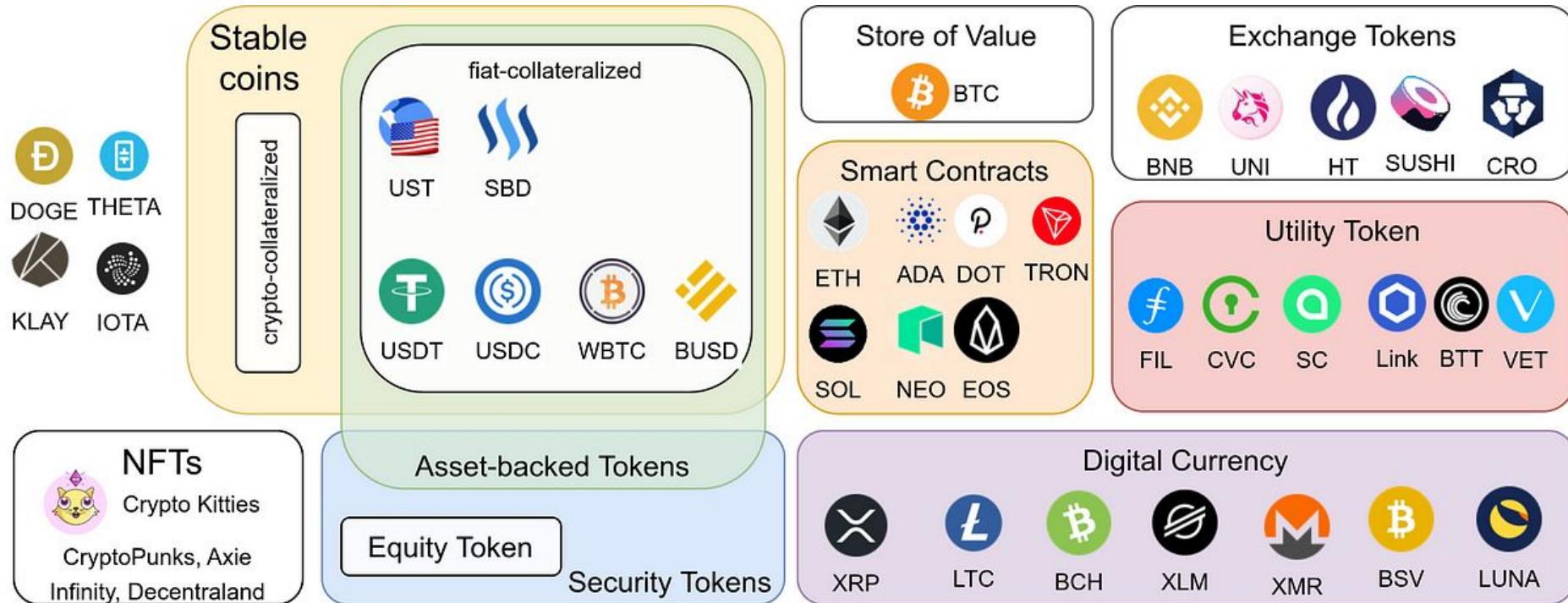


# 3. Cryptocurrency Basics



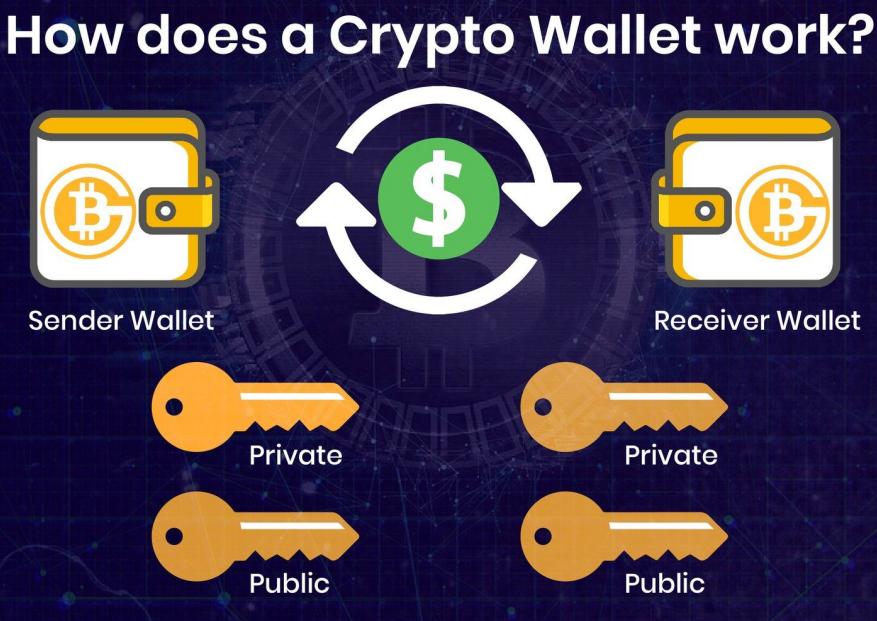


### 3. Cryptocurrency Basics



### 3. Cryptocurrency Basics

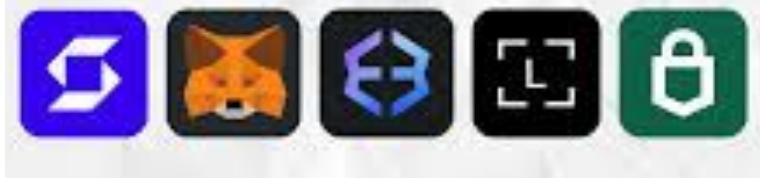
- **Cryptocurrency wallet** is a **software program that stores your digital money.**
- **store your private keys** - the passwords that give you access to your cryptocurrencies
- Keeps the crypto **safe and accessible.**
- **allow the user to send, receive, and spend cryptocurrencies** like Bitcoin and Ethereum.

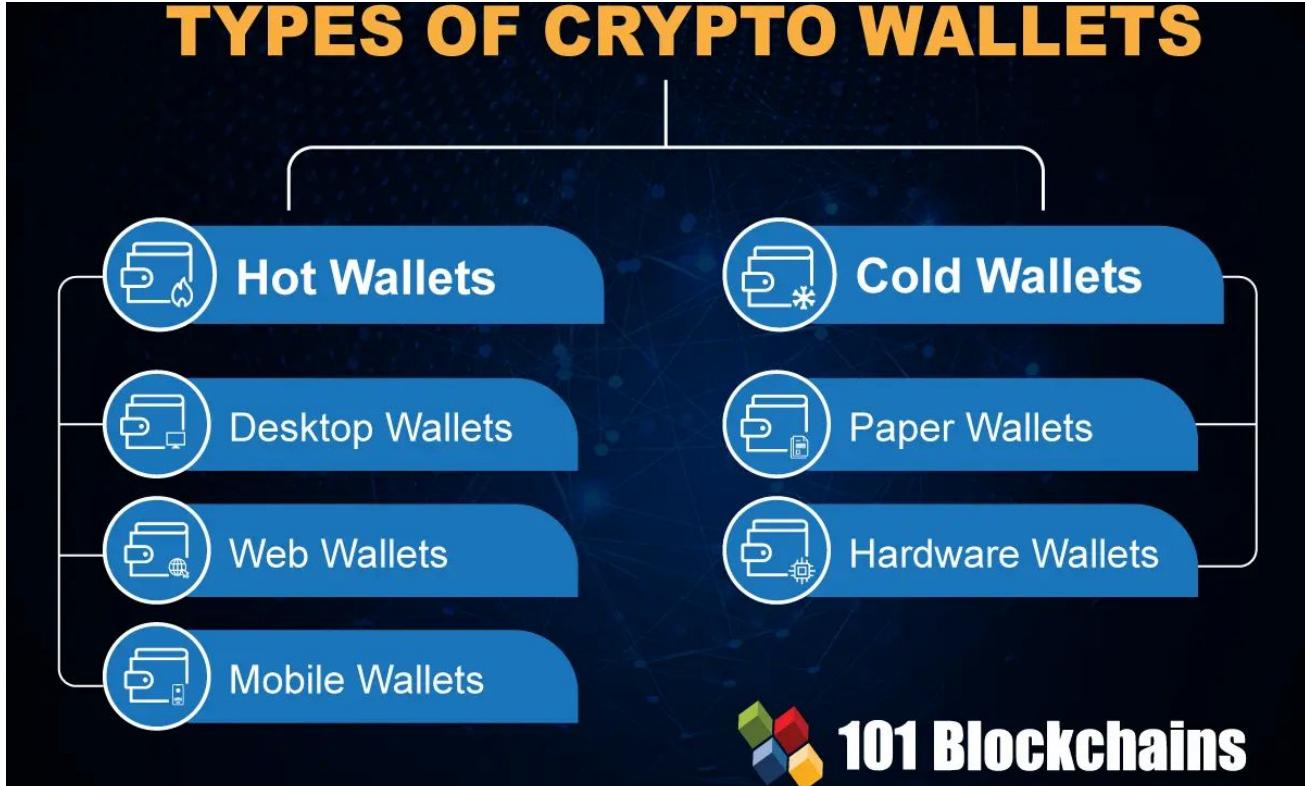


### 3. Cryptocurrency Basics

#### Why are crypto wallets important?

- Normal wallet - holds actual cash,
- Crypto wallets
  - **don't store your crypto** instead **store private keys**
    - used to access live holdings on the blockchain
    - prove the ownership of your digital money
    - Allows to perform transactions.
    - Losing private keys ⇒ losing access to your money.
  - That's why it's **important to keep the hardware wallet safe, or use a trusted wallet provider like Coinbase.**





### 3. Cryptocurrency Basics



### 3. Cryptocurrency Basics

Aspect	Security Token	Utility Token	Asset Token	Government Token (CBDC)
Primary Purpose	Represents ownership in assets/securities like equity or debt	Provides access to platform services/products	Digitally represents real-world assets (e.g., real estate, commodities)	Central bank-issued digital fiat for payments and monetary policy
Regulation	Treated as securities; SEC/equivalent oversight, KYC/AML required	Often unregulated if not investment contracts; platform-specific rules	Varies; often security-like if fractionalized assets	Fully regulated by central banks/governments
Value Source	Tied to underlying asset performance/dividends	Derived from platform utility/demand	Pegged/mirrors physical asset value	Backed by government/fiat reserves
Transferability	Restricted; secondary markets with compliance	Freely tradable on exchanges	Tradable but may have lockups	Controlled by central bank; programmable limits
Examples	tZERO Polymath (security tokenized stocks)	ETH (network access), BNB (exchange fees)	RealT (property shares), PAXG (gold)	China's e-CNY, Digital Euro pilots
Risk Profile	Investment risk (market/issuer)	Platform adoption risk	Asset volatility + smart contract risk	Sovereign risk; low volatility

# Topics to be covered

1. Introduction to Blockchain
2. Blockchain Fundamentals
3. Cryptocurrency Basics
4. **Introduction to Solidity Programming**





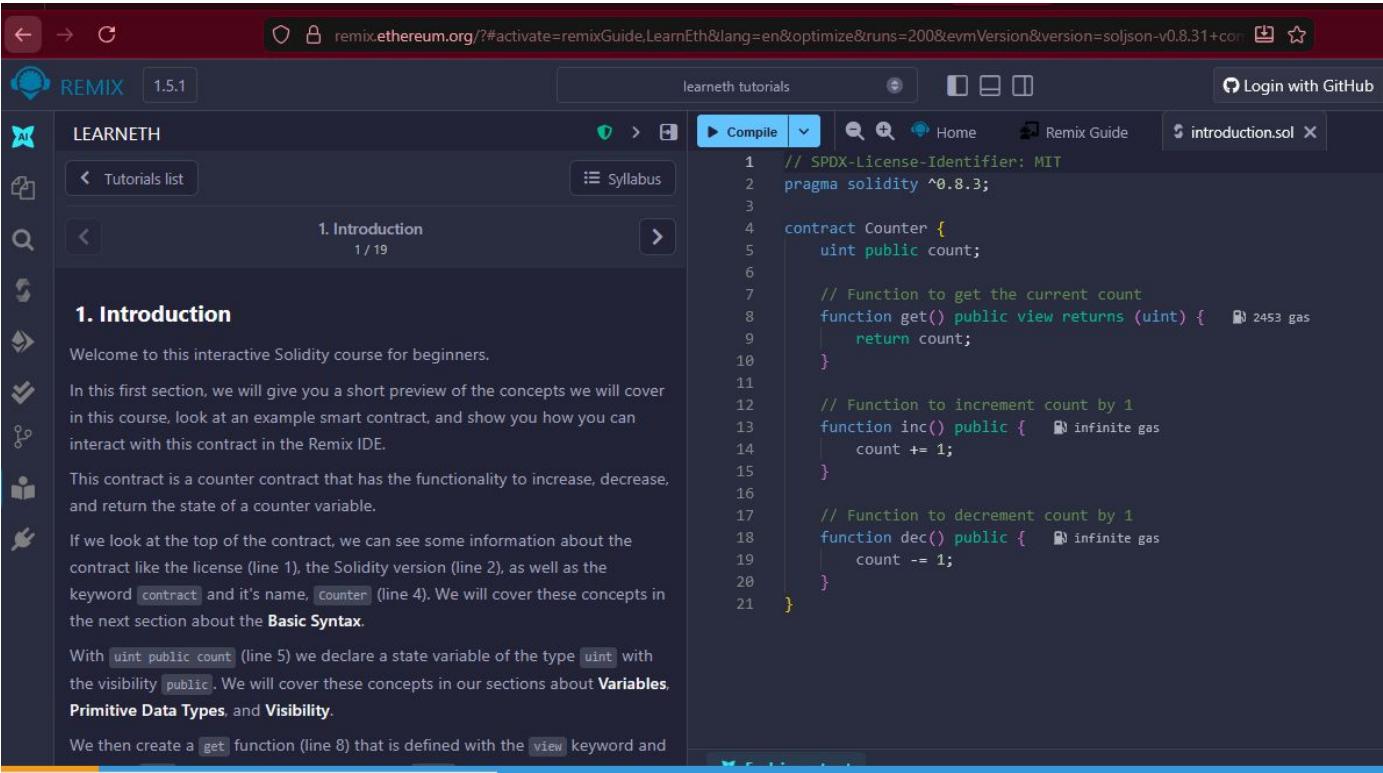
## 4. Introduction to Solidity Programming



- Variables - Local, Global, Variable, State Scope
- Data Types
- Operators - Arithmetic , Relational , Logical , Bitwise , Assignment , Conditional
- Functions
- Loops, if else
- Constructor, Visibility Modifier

# 4. Introduction to Solidity Programming

## LearnETH Platform (RemixIDE) - Beginners Course



The screenshot shows the LearnETH platform running in the Remix IDE. The left sidebar has icons for Tutorials list, Syllabus, and other course materials. The main content area is titled '1. Introduction' (1 / 19). The text says:

Welcome to this interactive Solidity course for beginners.

In this first section, we will give you a short preview of the concepts we will cover in this course, look at an example smart contract, and show you how you can interact with this contract in the Remix IDE.

This contract is a counter contract that has the functionality to increase, decrease, and return the state of a counter variable.

If we look at the top of the contract, we can see some information about the contract like the license (line 1), the Solidity version (line 2), as well as the keyword `contract` and its name, `Counter` (line 4). We will cover these concepts in the next section about the **Basic Syntax**.

With `uint public count` (line 5) we declare a state variable of the type `uint` with the visibility `public`. We will cover these concepts in our sections about **Variables**, **Primitive Data Types**, and **Visibility**.

We then create a `get` function (line 8) that is defined with the `view` keyword and

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract Counter {
    uint public count;

    // Function to get the current count
    function get() public view returns (uint) { 2453 gas
        return count;
    }

    // Function to increment count by 1
    function inc() public { infinite gas
        count += 1;
    }

    // Function to decrement count by 1
    function dec() public { infinite gas
        count -= 1;
    }
}
```