

# CS4160 - Blockchain Engineering

## Introduction to Blockchain Technology

**Can Umut ILERI**  
TU Delft, IOTA Foundation  
2023

## Who is the inventor of blockchains?

The answer depends on how you define blockchain!

### **First view:**

Growing lists of records (blocks) that are securely linked together via cryptographic hashes.

### **Second view:**

Technology that provides coordination between many parties, when there is no single trusted party.

A blockchain is a trusted coordination mechanism.

# Overview of the Lecture

# Overview of the Lecture

- Pre-Nakamoto: Blockchain as a Data Structure
- Nakamoto Consensus
- Post-Nakamoto:

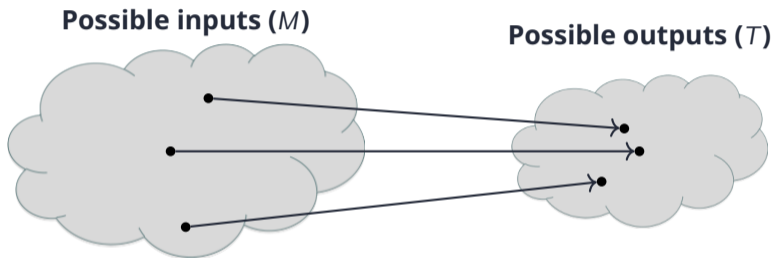
First, a bit of background.

# Cryptographic Fundamentals

# Cryptographic Fundamentals

- Hashing
- Asymmetric key encryption
- Digital dignatures

# Hashing



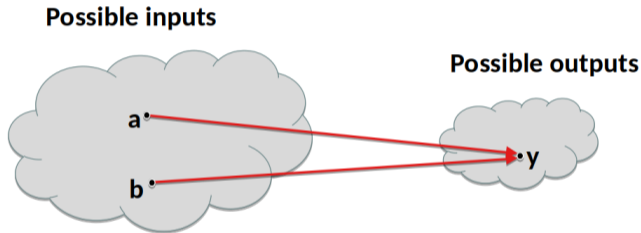
**Cryptographic hash function:** An efficiently computable function  $H : M \rightarrow T$  where  $|M| \gg |T|$ .

# Hashing

## Property 1: Collision-resistance

### Collision-resistance:

Collisions do exist but it is very difficult to find them



Finding two inputs  $a$  and  $b$  giving the same output  $y$  is infeasible.

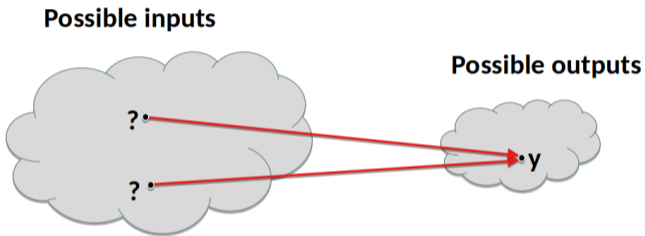


# Hashing

## Property 2: Hiding

### Hiding

Given an output, it is not feasible to find an input.



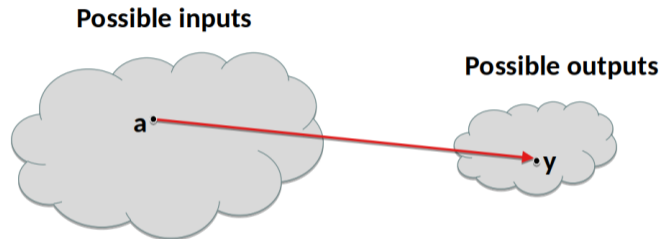
Given  $y$ , it is infeasible to find an input  $a$  such that  $H(a) = y$ .

# Hashing

## Property 3: Puzzle-friendliness

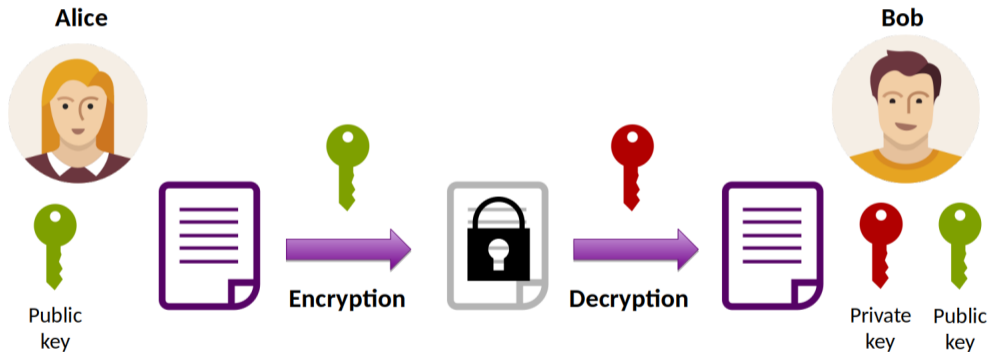
### Puzzle-friendliness

Given an output and only a part of the input, it is not feasible to find remaining part of input.

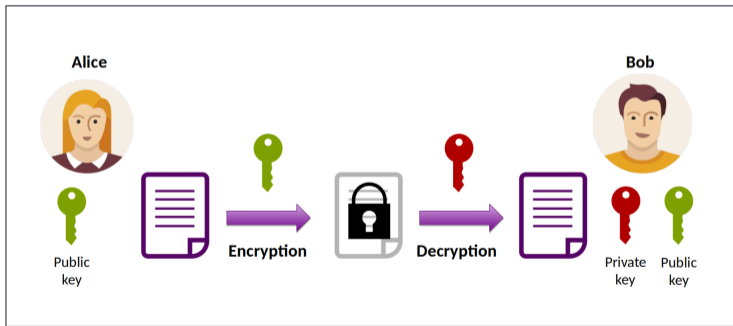


$a = 0010101101010?101011$        $y = 11100110$   
Given  $y$ , and all but one bit of its input  $a$ , there is no feasible way to predict the missing bit.

# Asymmetric Key Encryption



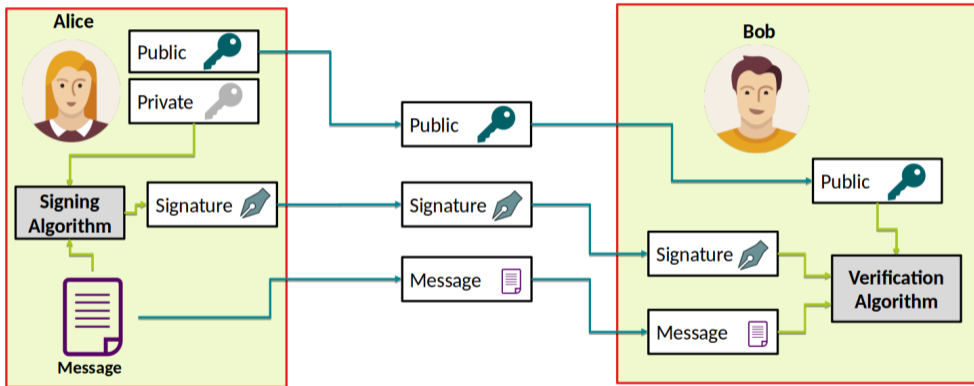
# Asymmetric Key Encryption



Charlie cannot decrypt Alice's message.

Charlie can still encrypt another message and send it to Bob.

# Digital Signatures



# The Blockchain Technology

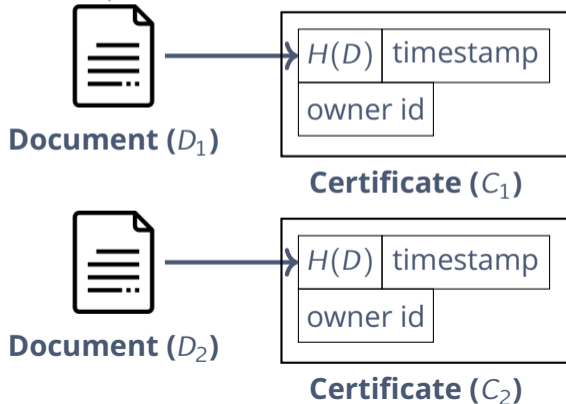
# Timestamping Digital Documents

[Haber and Stornetta, 1991]

- Documents are being digital.
- We need to
  - ▶ preserve and authenticate primacy of discovery,
  - ▶ prove that the history is not tampered.
- A naive solution: Trusted party that stores a copy of the document, together with a timestamp.
- Problem:
  - ▶ Privacy
  - ▶ Bandwidth and storage
  - ▶ Corrupted and lost documents
  - ▶ Malicious update of the time-stamp

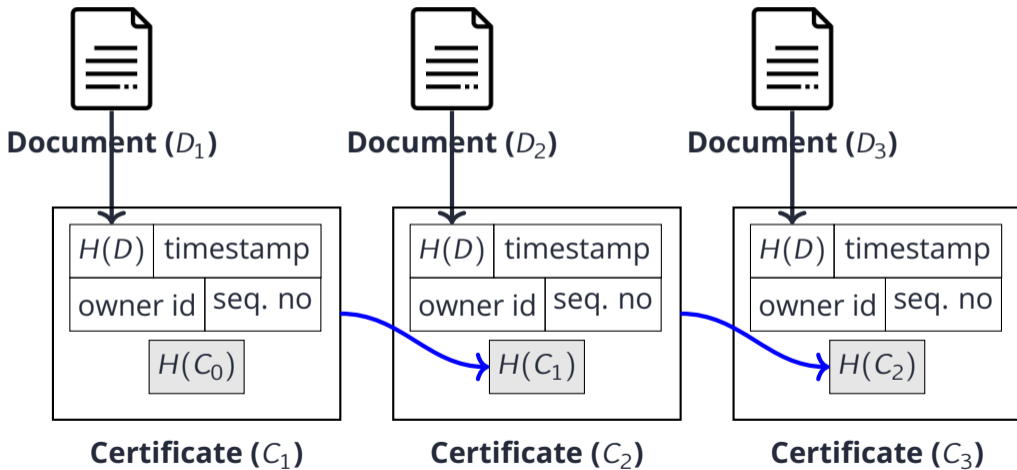
# Improving the Naive Approach

- Document provided with a hash
- Signature by trusted service, which shows:
  - ▶ the request was processed correctly,
  - ▶ hash and time stamp is correct.





# Hash chains



# Hash chains


Is everything perfect? Are all the documents/proofs in the safe hands? -  
Problem: single-point of failure.

**NOTICES &  
LOST AND  
FOUND**  
(5100-5102)

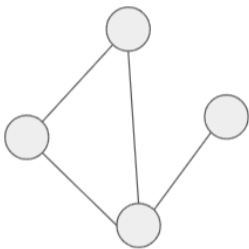
Universal Registry Entries:  
Zone 2 -  
dS8492cgVOFAoP9kyE1XzMOrQ  
HgEwzkVbVafNvikUz99avq8/ME  
p5y9EFG8XxzMBalGQQ==

Zone 3 -  
JnFCg+HCmvhj8GmmUP7VZna71  
NgZup/RfuKUQNzCHWXMuqLK  
durxHQV5pSHLaBGPRly+mg==

These base64-encoded values represent the combined fingerprints of all digital records notarized by Surety between 2009-06-03Z 2009-06-09Z.  
www.surety.com 571-748-5800

 One solution: Gain more trust.  
Another solution: Decentralization of trust!

# State machine replication



Three concerns:

- **Disseminating the command**
  - ▶ Fault-tolerant broadcast
- **Committing the command**
  - ▶ Fault-tolerant agreement
- **Executing the command**
  - ▶ Deterministic execution

# Bitcoin

## Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto  
satoshin@gmx.com  
www.bitcoin.org

**Abstract.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

# Motivation

## Concerns on Fiat currency

- Issued by governments.
- Delay between transaction and settlement.
- Financial institutions serving as trusted third parties.
- The cost of mediation increases transaction costs.
- We need a mechanism to make payments over a communications channel without a trusted party.

## Solution:

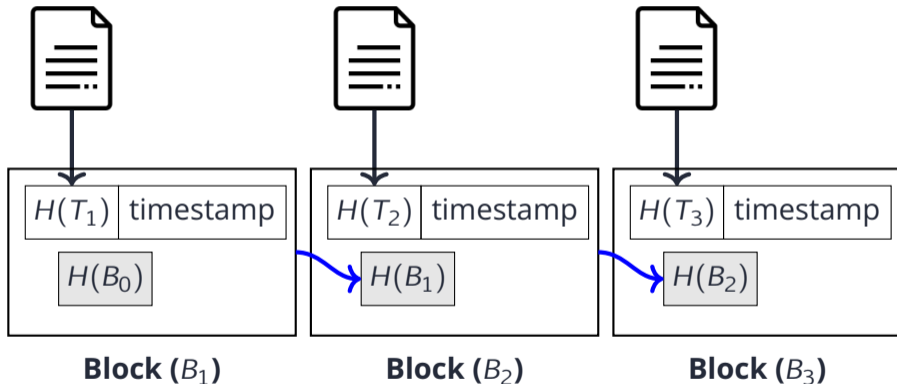
- A distributed ledger.

# Distributed Ledger

Transactions ( $T_1$ )

Transactions ( $T_2$ )

Transactions ( $T_3$ )



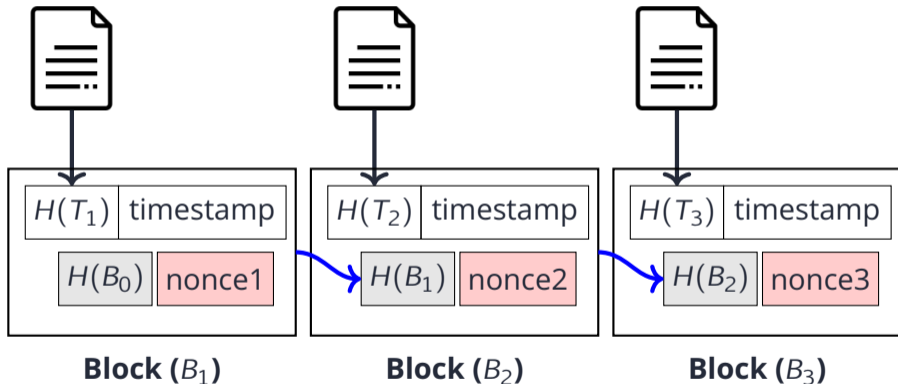
Who will determine the next block?

# Bitcoin Revolution: Nakamoto Consensus

Transactions ( $T_1$ )

Transactions ( $T_2$ )

Transactions ( $T_3$ )



Goal: find a  $nonce_t$  such that  $H(H(T_t), ts, H(B_{t-1}, nonce_t)) < d$

H: hash function

d: is the difficulty parameter



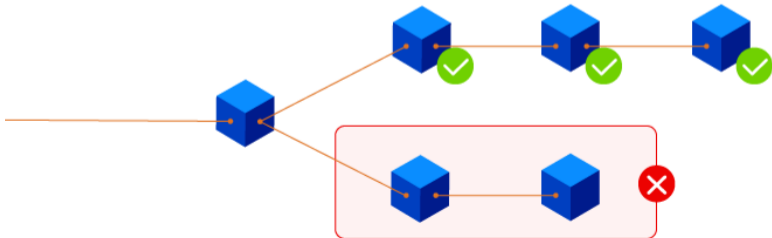
# Bitcoin Revolution: Incentives

- The first miner to find such a “magical” hash is rewarded with:
  1. The block reward (halves every 210,000 blocks, the coin reward decreased from 12.5 to 6.25 coins on 12 May 2020. It will decrease from 6.25 to 3.125 coins on May 6, 2024) (<https://www.bitcoinblockhalf.com>)
  2. Sum of transaction fees in the block (<https://billfodl.com/pages/bitcoinfees>) To get your transaction processed quickly you have to outbid other users
- Incentive mechanism
  - ▶ Transaction makers are incentivized to increase their transaction costs.
  - ▶ Nodes in the network are incentivized to be the one that finds the nonce.
  - ▶ Nodes have to use **scarce** resources to find the nonce.
  - ▶ <https://explorer.btc.com/stats/fee>

# Bitcoin

## Simultaneous Mining

- Bitcoin prevents **double-spending** during block and transaction validations.
- What if two miners find the same block at (roughly) the same time?
- Now, different miners will build upon different blocks. Double spending?
- Selection rule: longest chains wins.



# Bitcoin

- *Only 6 blocks or 1 hour is enough to make reversal computationally impractical.*
- Consistency: If no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.
- Is Bitcoin consistent?

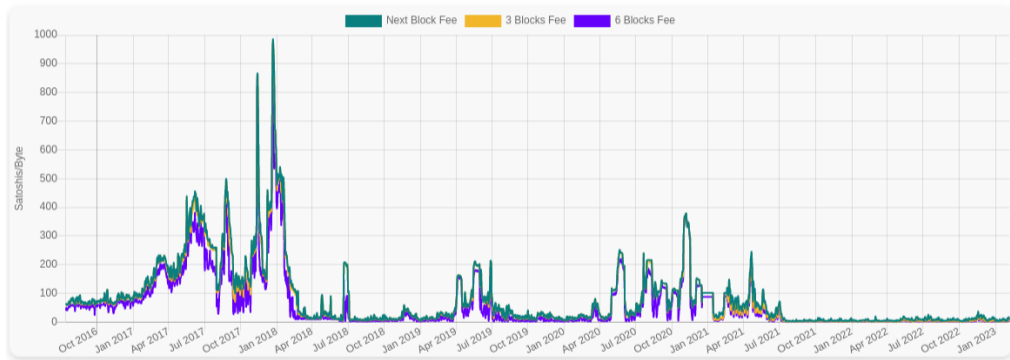
# Bitcoin

Adaptive race conditions:

- A block is expected to be found in 10 minutes.
  - ▶ If not, the difficulty is adjusted accordingly (after 2016 blocks).
  - ▶ The more miners are active, the more difficult the mining process becomes.
  - ▶ <https://explorer.btc.com/stats/diff>
- Bitcoin has fixed block sizes of 1MB.
  - ▶ Higher block size -> more transaction throughput.
  - ▶ But leads to faster growth of blockchain and resource usage.

# Bitcoin

## Transaction Costs



<https://privacypros.io/tools/bitcoin-fee-estimator/>

# Bitcoin

## Transaction Costs

### 200,000 Unconfirmed Transactions Pile Up in Another Crazy Day for Bitcoin



Exchanges overloaded. DDoS attacks. Huge price spikes, flash crashes, and order cancellations. Over 200k transactions queued, some for more than 24 hours, in the mempool – despite paying high fees. Just another crazy day in bitcoin, a land where every day seems to be wilder than the last, including higher highs, bigger swings, and record-breaking numbers across the board.

# Bitcoin

## Network and Nodes

### Bitcoin Full Nodes:

- download every block and transaction, and check them against rules:
  - ▶ validity of each transaction.
  - ▶ checking double spend.
  - ▶ signature check, block size check.
- Validated transactions are sent to *mempool*. <https://mempool.space/>
- Specialized full nodes:
  - ▶ **Pruned Full Nodes:** Given a storage limit, they only store latest blocks. Can still verify transactions, can download data, cannot upload data.
  - ▶ **Archival Full Nodes:** Stores a complete copy of the entire history of the Bitcoin blockchain from the genesis block. Used for queries.
  - ▶ **Super Nodes:** Large number of incoming and outgoing connections. Acts as a relay station and redistribution point.

# Bitcoin

## Network and Nodes

### Light Nodes:

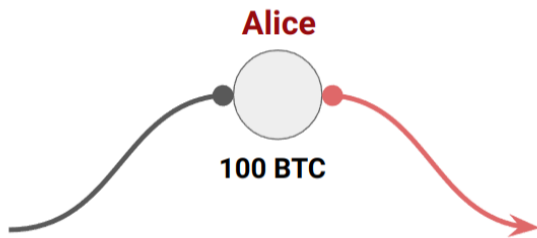
- They only download the block headers.
- have a similar role as pruned full nodes.
- use *simplified payment verification*(SPV) to verify transactions.
- cheaper to run and maintain.

### Miners:

- Select *attractive* transactions from the mempool.
- Compete to be the first to solve the block hash.

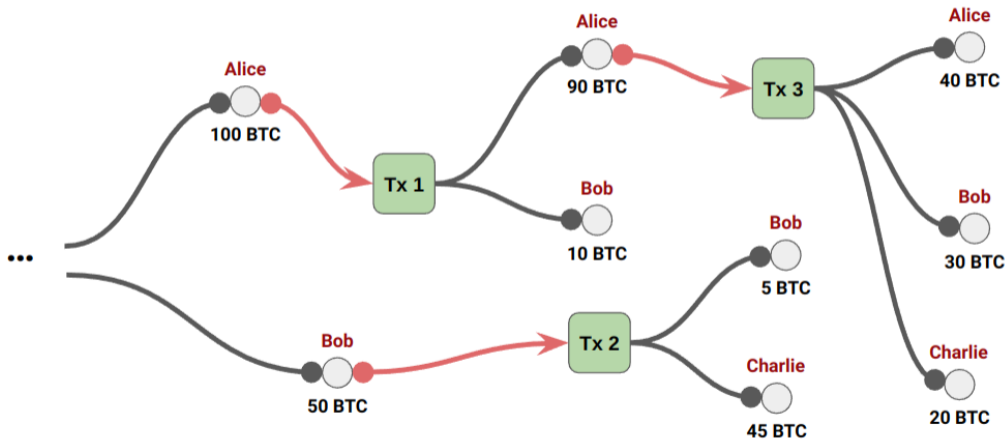


# Accounting in Bitcoin: UTXO Model

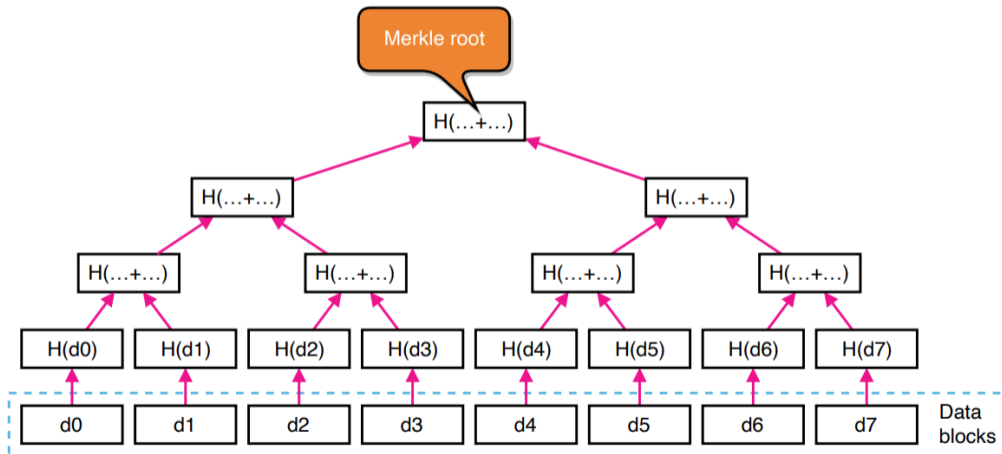


The UTXO holds a value of 100 BTC and it can only be spent if Alice permits.

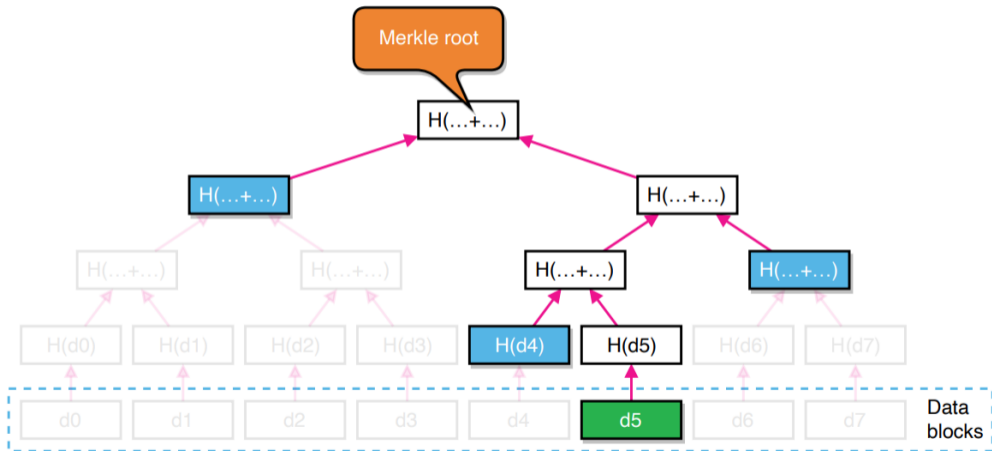
# Accounting in Bitcoin: UTXO Model



# Merkle Tree



# Merkle Tree



# Exploring Bitcoin

*DEMO*

<https://www.blockchain.com/explorer>

# Extending Blockchain Technology

# Extending Blockchain Technology

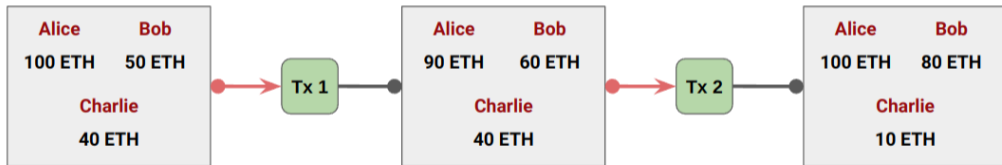
- Accounting
- Programmability
- Consensus
- Modularity
- Scalability
- Data structure
- Application

# Accounting



# Account-based accounting

- Represents coins as balances within an account.



# UTXO vs. Account Model

## The UTXO model:

- A **verification** model. Transactions are both **results** (of a local calculation) and **proofs**.
- It is enough to store txs. No need to deal with further with the states.
- TXs processed in parallel because they do not depend on any external state.

## The account model:

- A **computational** model: Users submit transactions instructing nodes on what state transitions should look like.
- The network then computes the new state based on the instructions.
- An account can be updated by more than one transactions within the same block.
- Program-friendly: UTXO's stateless model would force transactions to include state information.

# Programmability

# Programmability

## Ethereum Smart Contracts

- Smart contract: A collection of **code** (its functions) and **data** (its state) that resides at a specific **address**.
  - ▶ Smart contracts have balance, can receive and send assets.
  - ▶ Not controlled by a user. Instead, they are deployed to the network and run as programmed.
  - ▶ User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract.
  - ▶ Smart contracts are public on Ethereum and can be thought of as open APIs.
- Smart contracts extend the functionality of a simple transaction execution
  - ▶ When nodes handle your transaction, you guide them by deployed codes.
- Smart contracts are not self-executing. They have to be triggered by an on-chain transaction.

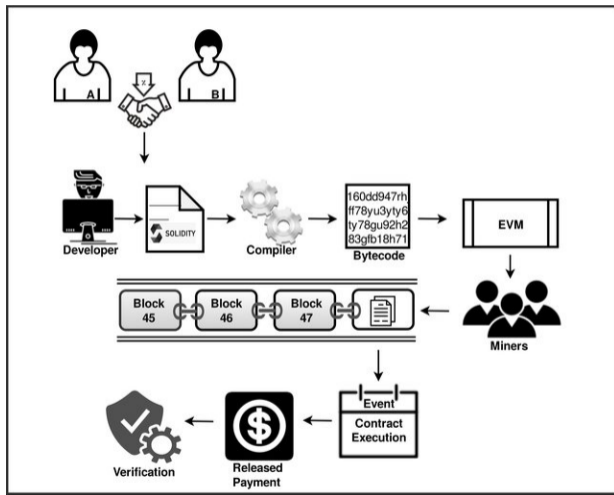
# Programmability

## Ethereum Smart Contracts

```
1 pragma solidity >=0.4.22 <0.7.0;
2 contract EtherBank {
3     mapping (address => uint256) public balances;
4     function deposit() external payable {
5         require(balances[msg.sender] + msg.value >= balances[msg.
6             sender]);
7         balances[msg.sender] += msg.value;
8     }
9     function withdraw(uint256 amount) external {
10        require(amount <= balances[msg.sender]);
11        balances[msg.sender] -= amount;
12        msg.sender.transfer(amount);
13    }
```

# Programmability

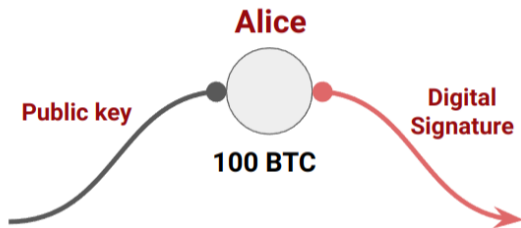
## Ethereum Smart Contracts



[Sayeed et al., 2020]

# Programmability

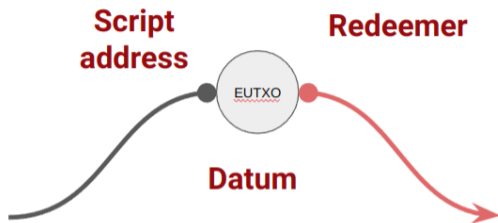
Cardano's eUTXO Model



Basic UTXO

# Programmability

## Cardano's eUTXO Model



**Script Address** :  $H(\text{BinaryOutput}(\text{PlutusSC}))$

**Redeemer**: User specific arguments

**Datum**: Arbitrary user data (Can be used as a local script state).

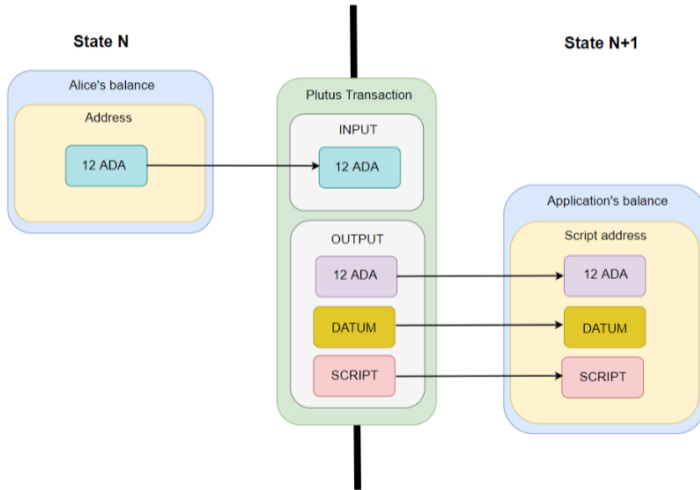
$\text{validator}(\text{Datum}, \text{Redeemer}, \text{ScriptContext}) \rightarrow \text{True}, \text{False}$

$\text{validator}(\text{LockerInput}, \text{UnlockerInput}, \text{ScriptContext}) \rightarrow \text{True}, \text{False}$



# Programmability

## Cardano's eUTXO Model



# Programmability

## Cardano's eUTXO Model

- Predictability:
  - ▶ If a transaction passes local validation, the user can be almost certain that the transaction will make it to a new block.
  - ▶ Same for Plutus scripts: if they pass the local check, fees most likely are not lost.
- One rule must be followed: *Each EUTXO can only be spent only once and as a whole within a block.*
- Advantages over Ethereum Smart Contracts:
  - ▶ It is possible to check whether a tx will validate in your wallet before you send it to the chain.
  - ▶ Much more limited scope. It is easier to understand what the script is doing what can possibly go wrong

# Programmability

## Web Assembly

- A binary instruction format for a stack-based virtual machine.
- Designed as a **portable** compilation target for programming languages.
- Enabling deployment on the web for client and server applications.
- General VM, contrary to EVM.
- Supports many languages (Rust, C/C++, C#, etc.).

# Programmability

## Move VM

- A new smart contract programming language with an emphasis on safety and flexibility.
- Writing secure code is very important. If you make a mistake in writing an application (e.g., a bug), you can easily end up blocking millions of dollars.
- Solidity is prone to errors.
- Avoid mistakes in the low-level code.
- Focus on how states and transactions are represented. If they are carefully designed, execution would be deterministic.
- Address some properties of physical assets that make them difficult to represent digitally.
  - ▶ Scarcity
  - ▶ Access control

# Consensus

# Proof of Stake

- Nodes can participate in consensus by *staking* and running software.
- One node with state is selected as block proposer for a time slot (12 seconds) with respect to
  - ▶ randomization,
  - ▶ staking age,
  - ▶ node's stake.
- Blocks are not *mined*, but *forged*. Block creator takes transaction fees.
- Benefits:
  - ▶ Less wasteful, environment-friendly.
  - ▶ Accessible. It does not require expensive equipment for mining.
    - As more nodes join, the system becomes more decentralized.
  - ▶ Security: Less attractive 51% attack.

# Ethereum's Proof of Stake

- A user creates and signs a transaction with their private key, determines fees and tips and sends to execution client.
- Execution client verifies validity (sufficient assets, correct signature).
- Execution client adds the transaction to its mempool, and gossips it.
- One of the execution clients is the block proposer for the current slot.
  - ▶ Execute transactions and compute state change.
  - ▶ Pass it to the consensus client.
- The node broadcasts the beacon block on the consensus layer network.
- Consensus clients re-execute the block and check validity of global state.

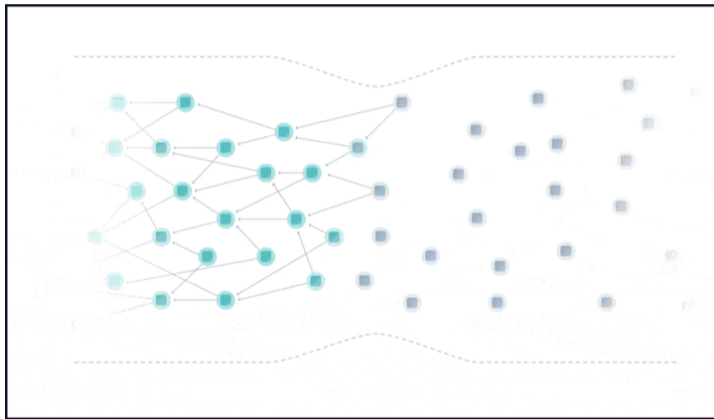
# Permissioned Blockchains

- Hyperledger Fabric
  - ▶ relies on a backend service (known as the ordering service) that intermediates the messages between senders and receivers.
  - ▶ backend service ensures that all receivers will see messages in same order.
- Ripple
  - ▶ XRP transactions are performed across a permissioned blockchain. Behind the blockchain network is a private company that runs a collection of private computer nodes that validate transactions.
  - ▶ Some of its partners: Bank of America, American Express, MoneyGram



# Data Structure

# Directed Acyclic Graphs



The Tangle

# Directed Acyclic Graphs

## IOTA: Tangle

- There is no single-chain of blocks.
- Each block verifies at least two other blocks.
- Parallel validation of transactions without requiring total ordering.
- No miners and validators → no transaction fees.
- Conflicts are resolved by
  - In the current version, there is a central coordinator which issues milestone blocks that other nodes trust.
  - In IOTA 2.0 will be fully decentralized.
    - ▶ stake and reputation-based weight function for conflict resolution.
- Tangle visualizer: <https://explorer.iota.org/mainnet/visualizer/>

# Directed Acyclic Graphs

## Avalanche

- Nodes stake AVA to become a validator.
- Transactions are stored on a DAG.
- For conflicting transactions, nodes ask each other's opinions.
- Through a decision procedure called Snowball, a transaction is finalized.

# Modularity

# Modularity

## Blockchain Components

Consensus

Execution

Data availability

Application

Bitcoin is a **monolithic** blockchain that handles all of these components. Ethereum somehow deals with execution and consensus separately, but it is still monolithic.

# Scalability Solutions

- Consensus is difficult.
- Consensus is slow.
- Writing data on-chain is expensive.
- Changing the *state* is expensive.

We can move the execution off-chain and still make use of the security guarantees of blockchains.

# Off-chain scalability solutions

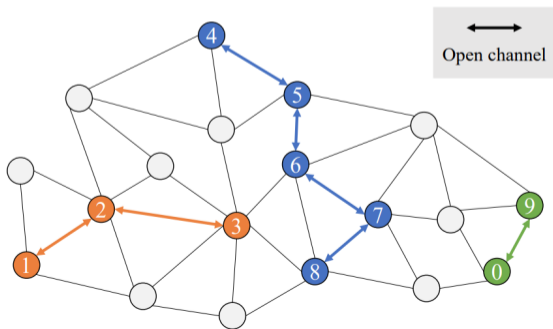
- Unidirectional payment channels:
  - ▶ Alice creates a transaction to send some coins to Bob.
  - ▶ She does not broadcast the transaction but sends it to Bob.
  - ▶ Bob can broadcast it whenever she wants.
  - ▶ Alice can create another transaction using the same UTXO.
- Bidirectional payment channels
  - ▶ Generalization of unidirectional payment channels.



# Off-chain scalability solutions

## Lightning network (on Bitcoin)

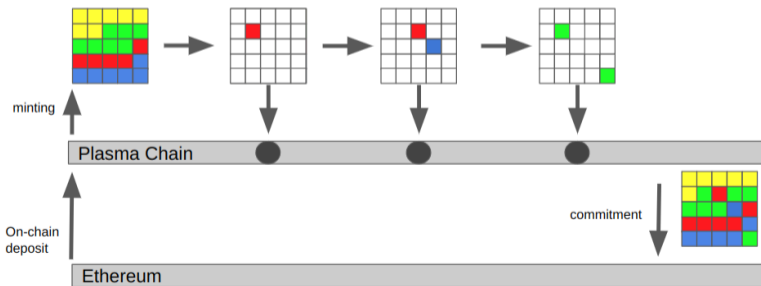
- Multisig transactions
- Both parties make a deposit into the channel to open a channel.
- They make trade on the channel, bypassing the blockchain.
- They close the channel and retrieve balances.



# Off-chain scalability solutions

## Plasma (on Ethereum)

- Make use of smart contracts.
- Data is shared with all users on Plasma.
- Dispute period / Fraud proofs.
- Data availability problem



# Off-chain scalability solutions

## Optimistic Rollups

- Coordinator collects user transactions, store and execute them locally (L2).
- Coordinator periodically submits Merkle tree root of transactions to L1.
- L1 Smart contract accepts commitments optimistically.
- Coordinator submits a compressed for of transaction data on L1.
  - ▶ They are not executed, but stored.
- Dispute resolution: Fraud proofs
  - ▶ L1 nodes can replay the transaction on-chain and compare.
  - ▶ If fraud proof is successful, cancel latest state transitions and slash sequencer's stake.
  - ▶ Everything older than the dispute period is final.

# Off-chain scalability solutions

## Validity-proof Rollups

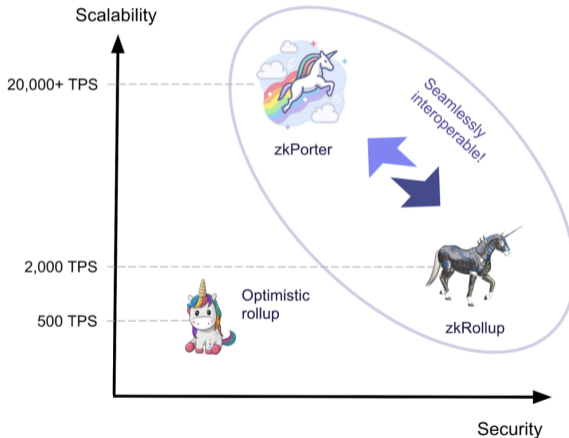
- Succinct zero-knowledge proofs for all state transitions.
- The zk-proof is committed alongside the state hash.
- The L2 commitment is accepted only if validity proof is verified.
- Transactions are stored (not executed) on L1.

## Validium

- Zero-knowledge validity proof.
- Data is not stored on-chain.
- A data availability proof is provided with the validity proof.

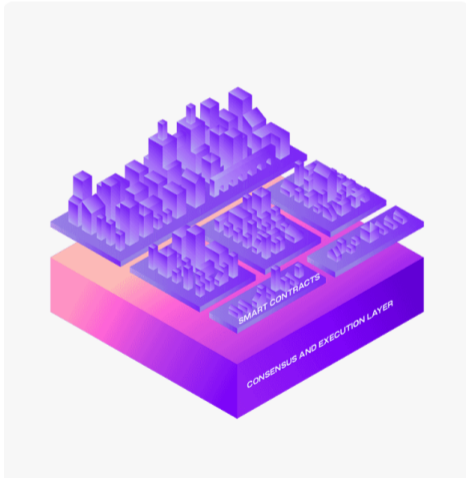
# Off-chain scalability solutions

## Validity-proof Rollups



# Data Availability Solutions

Celestia



# Data Availability Solutions

zk Rollup applications provide data availability services:

- MatterLabs: zkSync (Rollup) -> zkPorter (Data availability)
- StarkWare: StarkNet (Rollup) -> Volition (Data availability)
- Polygon: Hermez (Rollup) -> Avail (Data availability)

# Applications



# Applications

DeFi applications running on Ethereum

**Ethereum's DeFi**

The infographic is organized into 12 categories, each with a list of application logos and names:

- Payments:** request network, X Protocol, Dai Card, OPEN PLATFORM, xDai Chain, Groundhog, RAIDEN.
- Infrastructure:** connext, 0xcert, SETTLE, GITCOIN, DutchX, Ethlance, 0x, FOAM NETWORK, Bounties.
- KYC & Identity:** SELFKEY, sovrin, JOLOCOM, civic, uport, Bloom.
- Stablecoins:** SYNTHETIX, digix, DAI, USD Coin, GEMINI dollar, StableUnit, PAXOS STANDARD, TrueUSD, CARBON Reserve, Terra, Ampleforth.
- Insurance:** ETHERISC, Nexus Mutual, iXledger, VouchForMe, ai gang.
- Exchanges & Liquidity:** Eth 2.0 Dai, Centrifuge, AIRSWAP, Uniswap, kyber network, ForkDelta, Marble, IDEX, slow.trade, RADAR, ETHFINEX, TOTLE, hydro, LOOPRING, PARADEX, Bancor, Ren.
- Derivatives:** MARKETPROTOCOL, expo, LMM, veil, LENDROID, SY/SX, DAXIA, b2x, VARIABL.
- Credit & Lending:** LENDROID, Lendit, Compound, Celsius, Ripio Credit Network.
- Custodial Services:** MyEtherWallet, ZERION, argent, TRUST WALLET, METAMASK, Balance, MyCrypto.
- Investing:** Set, HARBOR, 22X, SWARM, FETCH, MELONPORT, Brickblock, SPiCE, bskt, MERIDIO, BETOKEN, SLICE, SCIENCE BLOCKCHAIN, MATTEREUM.
- Marketplaces:** Rare Bits, district0x, ORIGIN, OpenSea.
- Prediction Markets:** Guesser, augur, Bodhi, veill, STOX, GNOSIS.

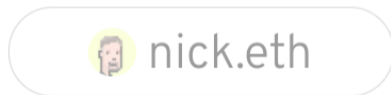
All codes are public.

# Applications

## Name Service

### Your web3 username

No more sandboxed usernames. Own your username, store an avatar and other profile data, and use it across services.



`https://ens.domains`

# Applications

## Decentralized Autonomous Organizations

A DAO is a collectively-owned, blockchain-governed organization working towards a shared mission.

- Work with like-minded folks around the globe without trusting a benevolent leader to manage the funds or operations.
- Blockchain-based rules baked into the code define how the organization works and how funds are spent. (No CEO)

# Applications

## Decentralized Autonomous Organizations

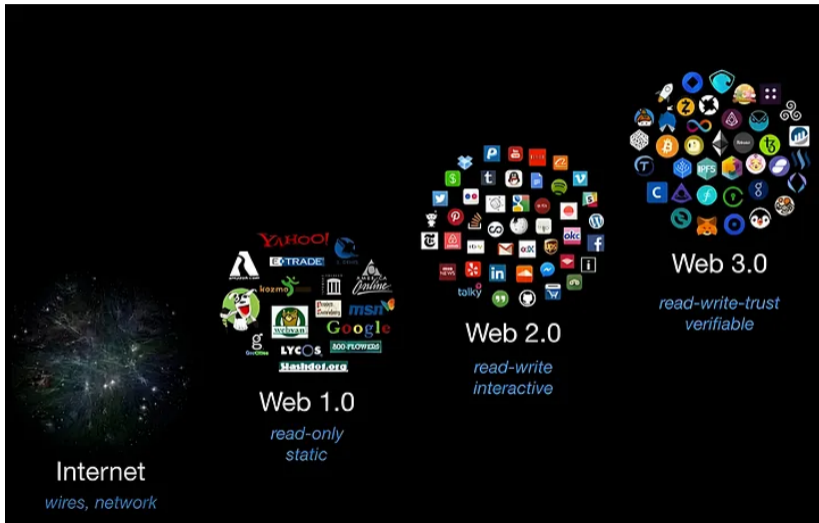
Join a DAO?

[https://ethereum.org/en/community/get-involved/  
#decentralized-autonomous-organizations-daos](https://ethereum.org/en/community/get-involved/#decentralized-autonomous-organizations-daos)

# Running applications on blockchain

- Transparency
  - ▶ Open-source.
  - ▶ Nobody says we cannot tell you what we are doing.
  - ▶ Everybody publishes as fast as they can.
- Public-verifiability
  - ▶ In the real world, we trust banks, governments, companies.
  - ▶ In blockchains everything is publicly verifiable.
-

# Web3



# Web3

## Core ideas of Web3

- Web3 is **decentralized**: ownership gets distributed amongst its builders and users.
- Web3 is **permissionless**: everyone has equal access to participate in Web3.
- Web3 has **native payments**: it uses cryptocurrency for spending and sending money online.
- Web3 is **trustless**: it operates using incentives and economic mechanisms.

# References I



Haber, S. and Stornetta, W. S. (1991).

How to time-stamp a digital document.

*Journal of Cryptology*, 3:99–111.



Liu, C., Gao, J., Li, Y., Wang, H., and Chen, Z. (2020).

Studying gas exceptions in blockchain-based cloud applications.

*Journal of Cloud Computing*, 9.



Sayeed, S., Marco-Gisbert, H., and Caira, T. (2020).

Smart contract: Attacks and protections.

*IEEE Access*, PP:1–1.