

# **Innovative Trends: Blockchain**

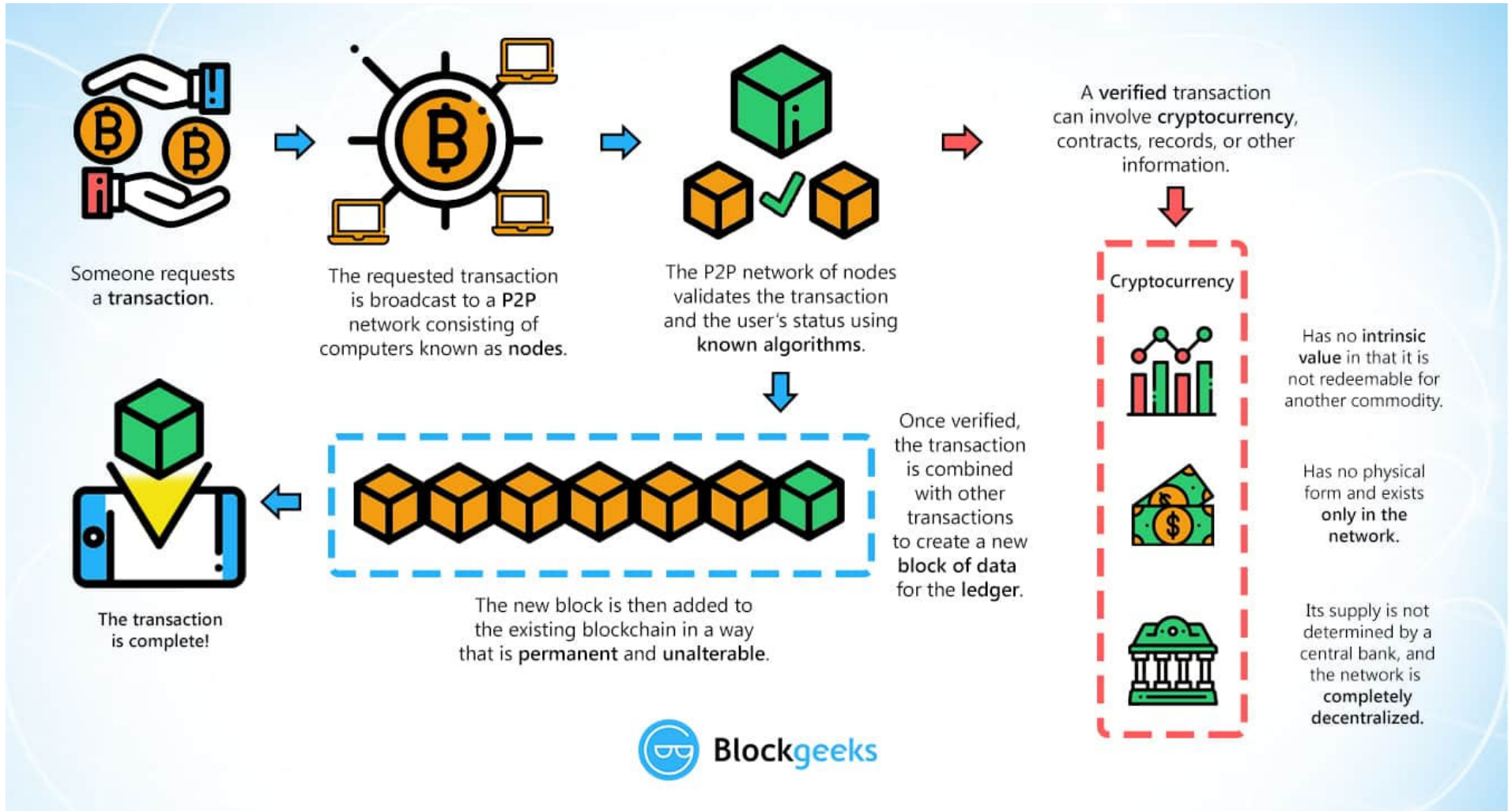
**Andrei Vasilățeanu**

# Outline

- **Introduction**
- Blockchain properties
- Bitcoin transactions
- Consensus
- Types of blockchains
- Merkle trees
- Smart contracts
- dApps

# What is blockchain

- *"The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value."*
- A time-stamped series of immutable records of data that is managed by a cluster of computers not owned by any single entity
- Each of these blocks of data (i.e. block) is secured and bound to each other using cryptographic principles (i.e. chain).



# Some history

- transacting money/value -> intermediaries like banks and governments
- need for intermediaries increases in case of digital assets which are easy to reproduce:
- double spending problem (act of spending the same unit of value more than once )

# Outline

- Introduction
- **Blockchain properties**
- Bitcoin transactions
- Consensus
- Types of blockchains
- Merkle trees
- Smart contracts
- dApps

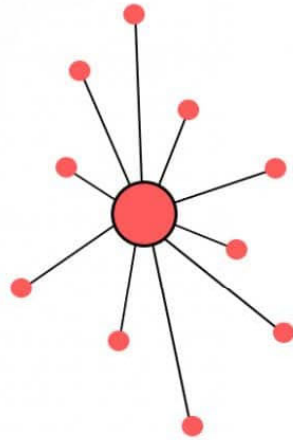
# Blockchain properties

- no central authority
- shared and immutable ledger -> network of replicated databases, synchronized via the internet and visible to anyone within the network
- transparent
- no transaction cost – Bitcoin monetary transactions
- trustless system (actually no trust is required)
- can be generalized to replace all processes and business models that rely on charging a small fee for a transaction (think Uber, Amazon)

# Decentralization

## The New Networks

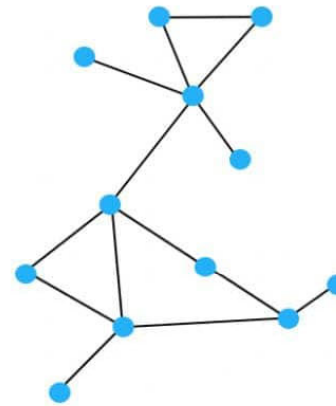
### Centralized



Centralized systems have a core authority that **dictates the truth** to the other participants in the network.

Only **priveleged users** or institutions can access the history of transactions or confirm new transactions.

### Decentralized



Decentralized systems have **no core authority** to dictate the truth to other participants in the network.

**Every participant** in the network can access the history of transactions or confirm new transactions.





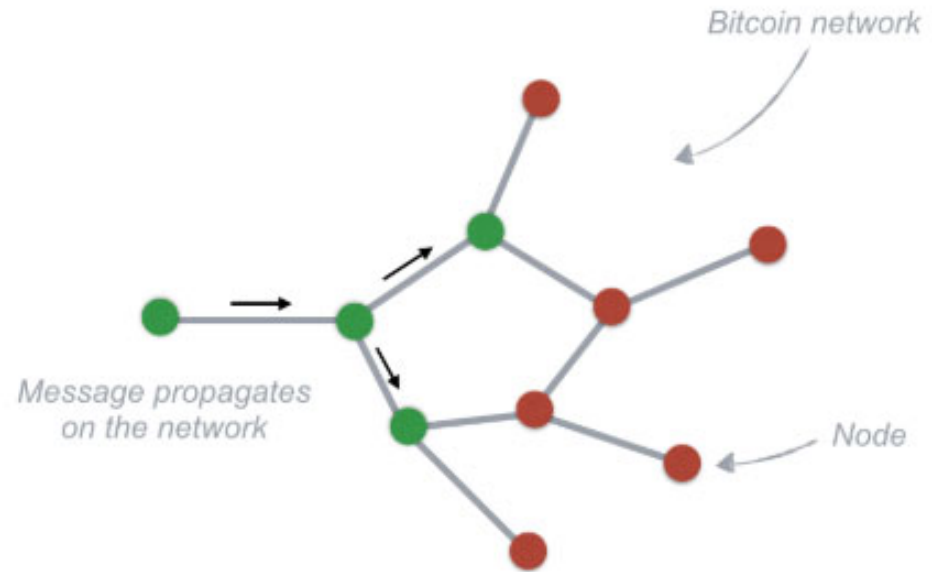


**LEDGER** ●

Account owner	Value
Mary	4
John	56
Sandra	83
Lisa	16
David	187
Brian	23

**LEDGER** ●







Account owner	Value
Mary	4
John	56
Sandra	88
Lisa	16
David	182
Brian	23



Each *node* receives the transaction request message, updates its own copy of the *ledger* and passes on the message to the nearby *nodes*.

# Transparency

- while identities can be secure, all transactions that were done by their public address are visible

TxHash	Block	Age	From	To	Value	[TxFee]
<a href="#">0x2d055e4585ae2a...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0x003e3655090890...</a>	 <a href="#">0x2bdc9191de5c1b...</a>	0,004741591554641 Ether	0.000294
<a href="#">0xb4d37c791ff4cde...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0x6c3b4faf413e0e4...</a>	 <a href="#">0xf14cb3acac7b230...</a>	0,744767225 Ether	0.000294
<a href="#">0x9979410dcb5f4c...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0x99bcd75abbac05...</a>	 <a href="#">0x2d42ee86390c59...</a>	0,016294 Ether	0.000294
<a href="#">0x189c4d4aae09be...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0x175cd602b2a1e7...</a>	 <a href="#">0xd39681bb0586fb...</a>	0,01 Ether	0.000294
<a href="#">0xda0e9bbb11fb77...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0x73a065367d111c...</a>	 <a href="#">0x01995786f14357...</a>	0 Ether	0.00150007
<a href="#">0x6be498fafad9acb...</a>	<a href="#">5629306</a>	16 secs ago	<a href="#">0xa3eb206871124a...</a>	 <a href="#">0x8a91cac422e55e...</a>	0,029594 Ether	0.000294

# Immutability

- Basically impossible to change previous blocks
- blockchain is a linked list that contains data and a hash pointer that points to its previous block
- hash pointer is similar to a pointer, but instead of just containing the address of the previous block it also contains the hash of the data inside the previous block  
=> changing previous blocks means changing all the next blocks

# Outline

- Introduction
- Blockchain properties
- **Bitcoin transactions**
- Consensus
- Types of blockchains
- Merkle trees
- Smart contracts
- dApps

# Bitcoin transactions

- Input and output:
- UNSPENT TRANSACTION OUTPUTS (UTXO)
- list of “unspent” Bitcoin amounts that have been sent *to* a user, but have not yet been sent *from* him/her
- collection of Bitcoin amounts on different addresses, and the role of a wallet is to identify which addresses the user has keys to
- A transaction is valid if one can prove ownership over the actual Bitcoin s/he is trying to send.
- similar to a cash economy

# Bitcoin transactions

- blockchain system doesn't keep track of account balances at all; it only records each and every transaction that is verified and approved
- to determine your wallet balance, you need to analyze and verify all the transactions that ever took place on the whole network connected to your wallet

LEDGER	
Transactions	Value
Mary → John	10.000
John → Lisa	0.345
Sandra → David	18.4332
Lisa → Sandra	7.156
David → Mary	12.3402
Brian → Lisa	3.029381
...	...

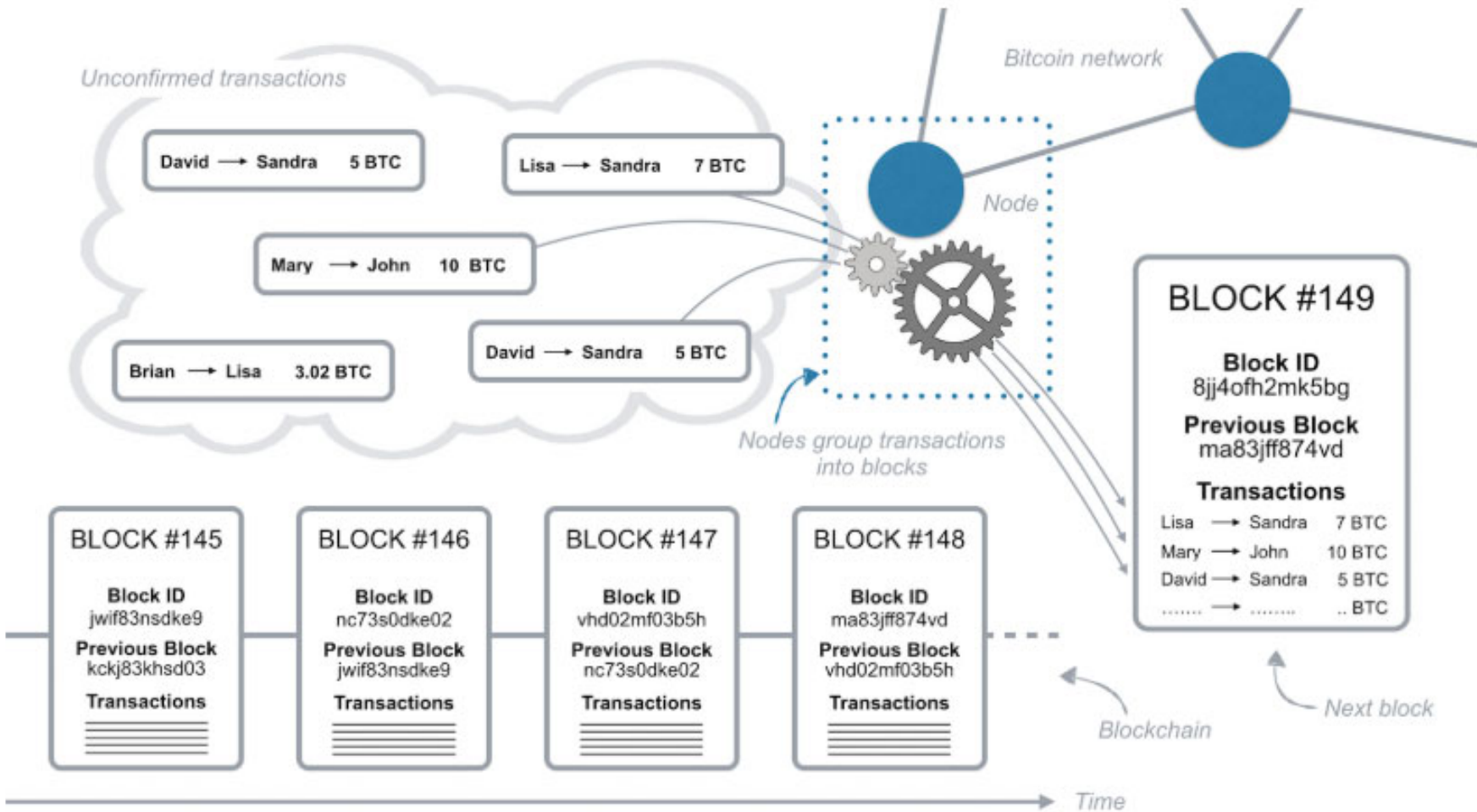
# Bitcoin transactions

- Valid transactions - check all the previous transactions correlated to the wallet you use to send bitcoins via the input references
- Owning bitcoins means that there are transactions written in the ledger that point to your wallet address and haven't been used as inputs yet

# Bitcoin transactions

- Signed piece of data that is broadcast to the network and, if valid, ends up in a block in the blockchain
- *Transfer ownership* of an amount of Bitcoin to a Bitcoin address
- When a digital transaction is carried out -grouped together in a cryptographically protected block with other transactions that have occurred in the last 10 minutes and sent out to the entire network.
- Miners then compete to validate the transactions by solving complex coded problems.





## Bitcoin Transaction Example

txid 90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219

```
{
  "hash": "90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219",
  "ver": 1,
  "vin_sz": 1,
  "vout_sz": 2,
  "lock_time": 0,
  "size": 226,
  "in": [
    {
      "prev_out": {
        "hash": "18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
        "n": 0
      },
      "scriptSig": "3045022100c1efcad5cdcc0dcf7c2a79d9e1566523af9c7229c78ef71ee8b6300ab...[snip]"
    }
  ],
  "out": [
    {
      "value": "5.93100000",
      "scriptPubKey": "OP_DUP OP_HASH160 4b358739fc7984b8101278988beba0cc00867adc OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value": "1678.06900000",
      "scriptPubKey": "OP_DUP OP_HASH160 55368b388ccfe22a3f837c9eee93d053460db339 OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
}
```

- tx format version - currently at version 1
- in-counter - number of input amounts
- out-counter - number of output amounts
- tx lock\_time - should be 0 or in the past for the tx to be valid and included in a block
- size - of the transaction in bytes

image by Venzen <venzen@mail.bihthai.net> 2014 CC SA  
conditions of reuse: <http://sofala.bihthai.net/works/txinout.htm>

### Two People Send Amounts To Addresses Held By a Bitcoin Wallet

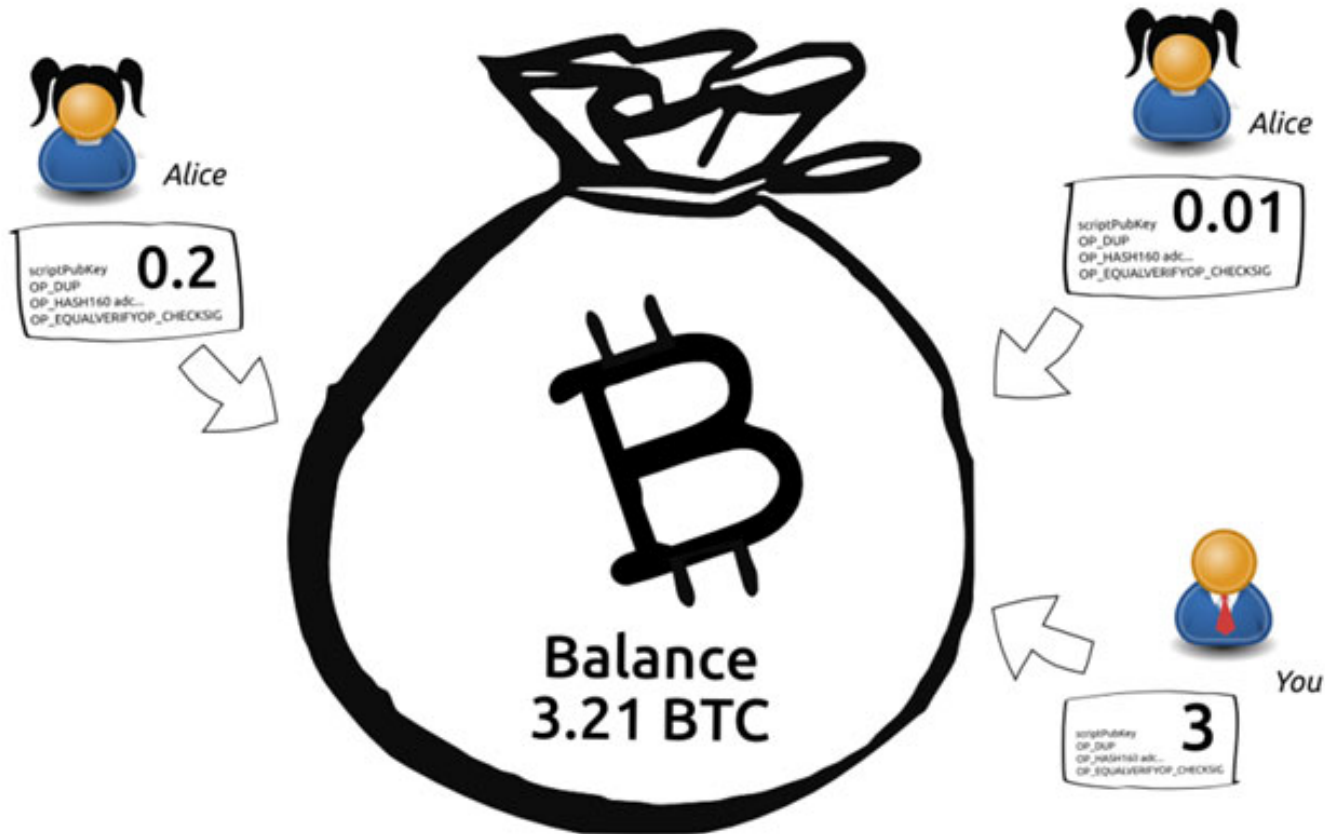


Image by Vincent vanoni@visualshock.net - 2014 CC SA  
conditions of reuse: <http://shutts.bitcoin.net/works/byvisual.htm>



*Spending Consumes UTXOs and Creates New Ones*

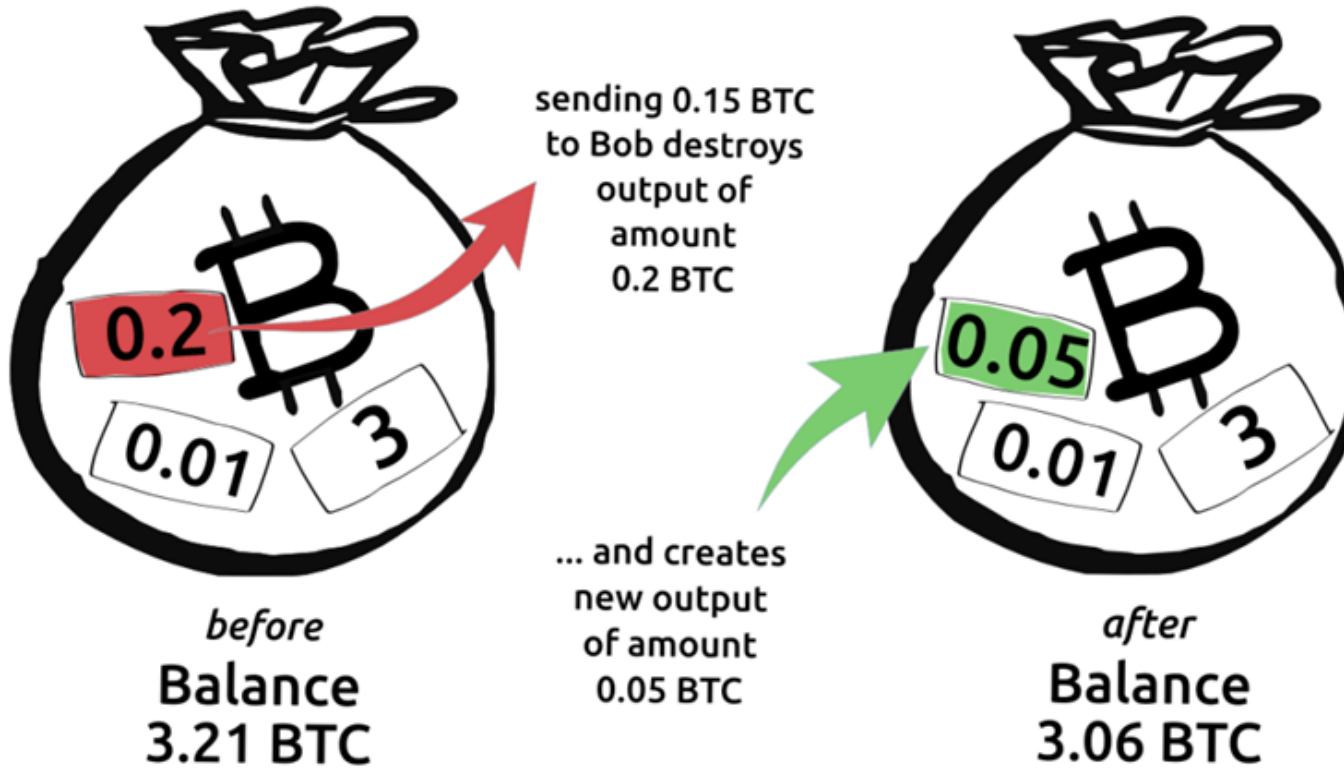


image by Venzen <venzen@mail.bitshai.net> 2014 CC SA  
conditions of reuse: <http://info.bitshai.net/works/txinout.htm>

# Outline

- Introduction
- Blockchain properties
- Bitcoin transactions
- **Consensus**
- Types of blockchains
- Merkle trees
- Smart contracts
- dApps

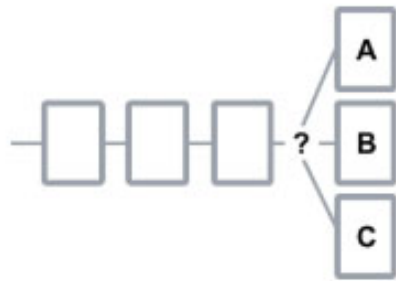
# Consensus mechanism

- confirm the transactions without relying on a trusted third party
- Proof of work
- transaction is initiated, the information is stored in a candidate block which is filled with the transaction's information
- miners get to work on solving a cryptographic puzzle that has a prize for whomever solves it, in the form of newly minted coins/currency.
- Proof of stake
- miners that have more "money", cryptocurrency, or "skin in the game" to have a greater opportunity to mine blocks and make decisions for the network

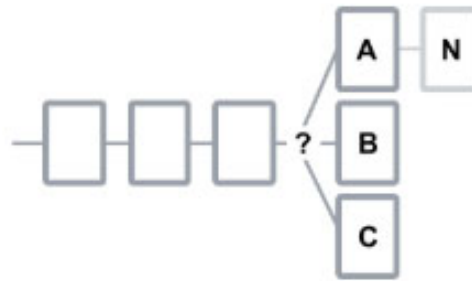
# Consensus

- Node that solves the mathematical problem acquires the right to place the next block on the chain and broadcast it to the network
- what if two nodes solve the problem at the same time and send their blocks to the network simultaneously?
- both blocks are broadcast and each node builds on the block that it received first
- blockchain system requires each node to build immediately on the longest blockchain available

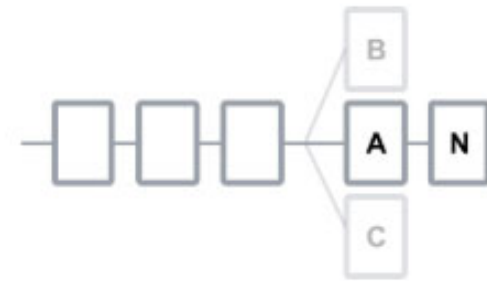




*Sometimes multiple nodes solve the mathematical problem at the same time generating end-of-chain ambiguity about what is the next block*

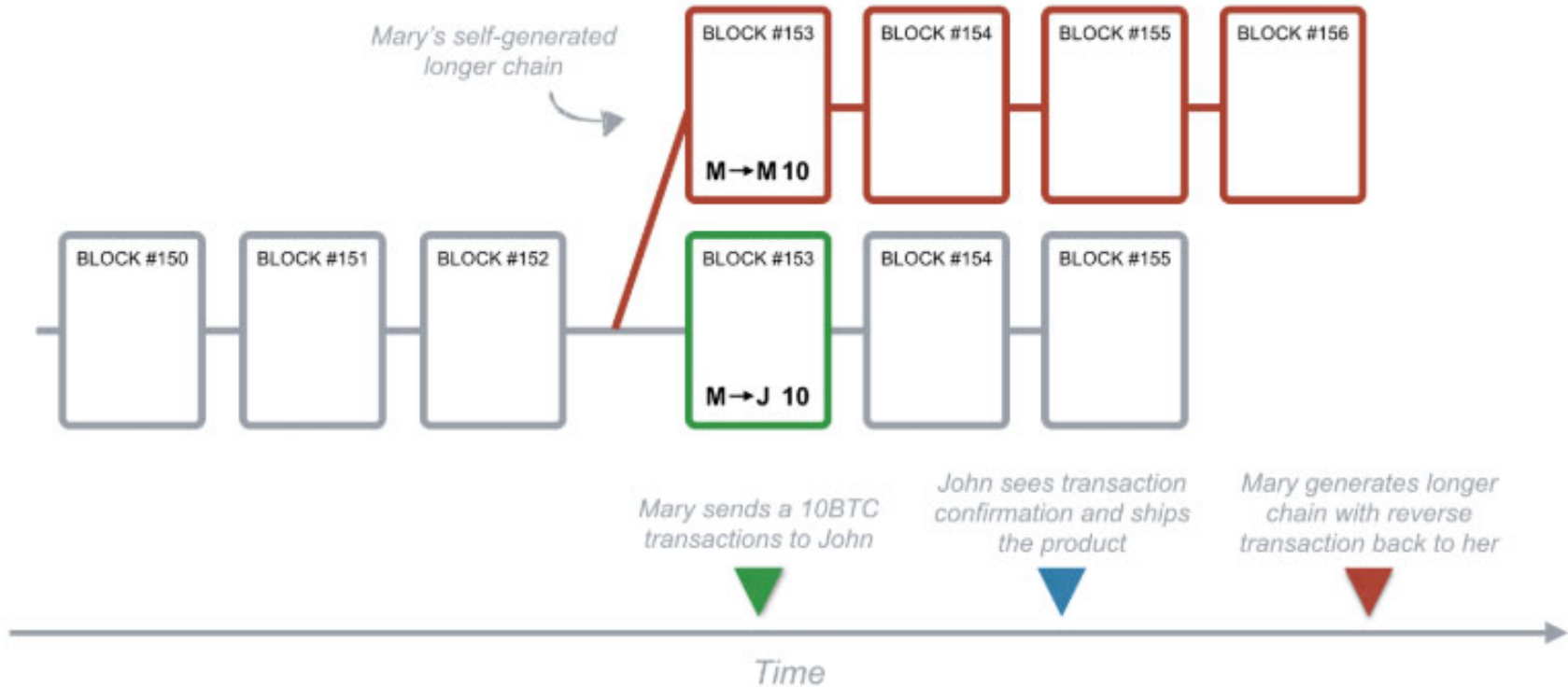


*Each node then tries to add the new block (N) to the block they received first from the other nodes*



*As soon as the new block (N) is added all the network adopt the longest chain possible (A+N) stabilising the whole network*

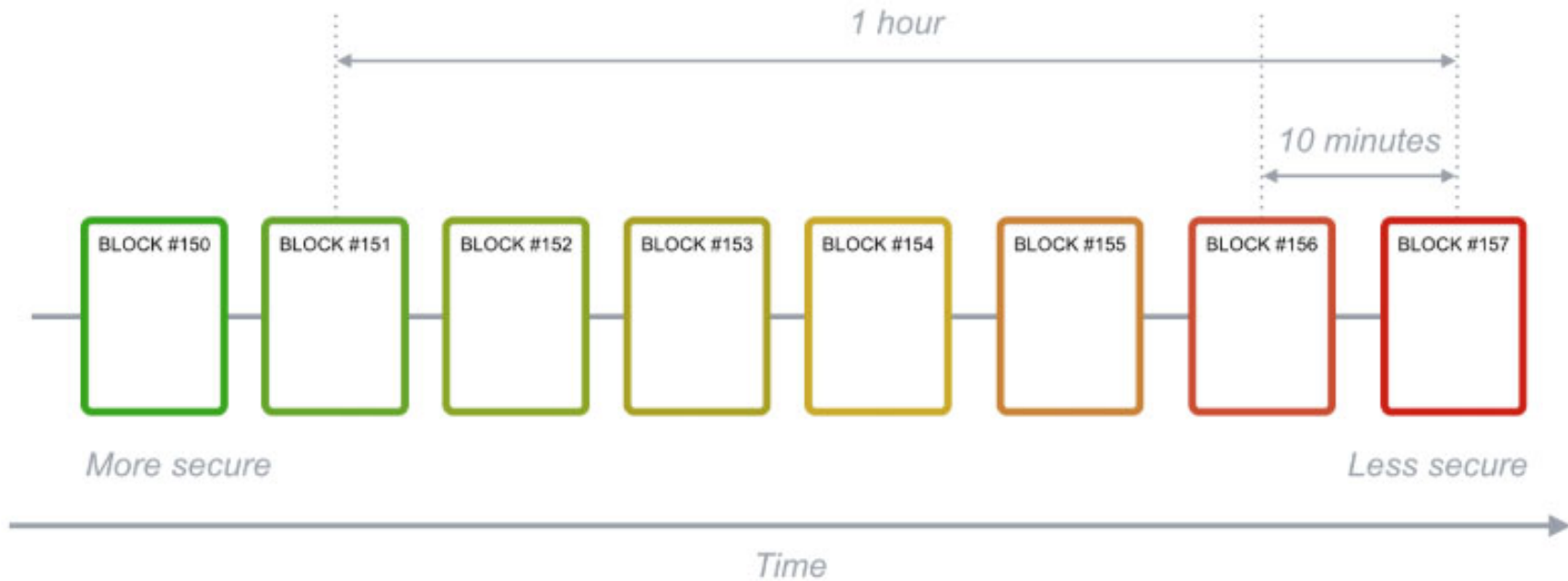
# Double spending attack



# Double spending attack

- Mary must generate a longer tail with the reverse transaction - race against the rest of the network to solve the math problem that allows her to place the next block on the chain
- Very unlikely to solve two, three, or more blocks in a row
- Even matching 50 percent of the computing power of the network – 25 percent chance for the second block in the row
- Difficulty of the problem increases with the size of the network

# Blockchain transactions security





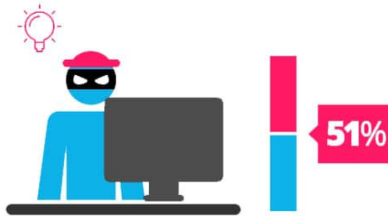
## Proof of Work

vs.

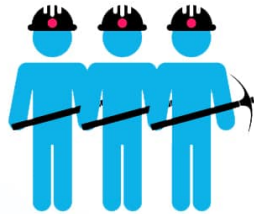
## Proof of Stake



To add each block to the chain, miners must compete to solve a difficult puzzle using their computers processing power.



In order to add a malicious block, you'd have to have a computer more powerful than 51% of the network.



The first miner to solve the puzzle is given a reward for their work.



There is no competition as the block creator is chosen by an algorithm based on the user's stake.



In order to add a malicious block, you'd have to own 51% of all the cryptocurrency on the network.



There is no reward for making a block, so the block creator takes a transaction fee.

# Outline

- Introduction
- Blockchain properties
- Bitcoin transactions
- Consensus
- **Types of blockchains**
- Merkle trees
- Smart contracts
- dApps

# Public, private blockchains

- **Public blockchains:** anyone in the world can read, anyone in the world can send transactions to and expect to see them included if they are valid, and anyone in the world can participate in the *consensus process* .
- are secured by **cryptoeconomics** - the combination of economic incentives and cryptographic verification using mechanisms such as proof of work or proof of stake, following a general principle that the degree to which someone can have an influence in the consensus process is proportional to the quantity of economic resources that they can bring to bear. "fully decentralized".

# Public, private blockchains

- **Consortium blockchains:** consensus process is controlled by a pre-selected set of nodes; for example, one might imagine a consortium of 15 financial institutions, each of which operates a node and of which 10 must sign every block in order for the block to be valid. The right to read the blockchain may be public, or restricted to the participants, and there are also hybrid routes such as the root hashes of the blocks being public together with an API that allows members of the public to make a limited number of queries and get back cryptographic proofs of some parts of the blockchain state.
- "partially decentralized".



# Public, private blockchains

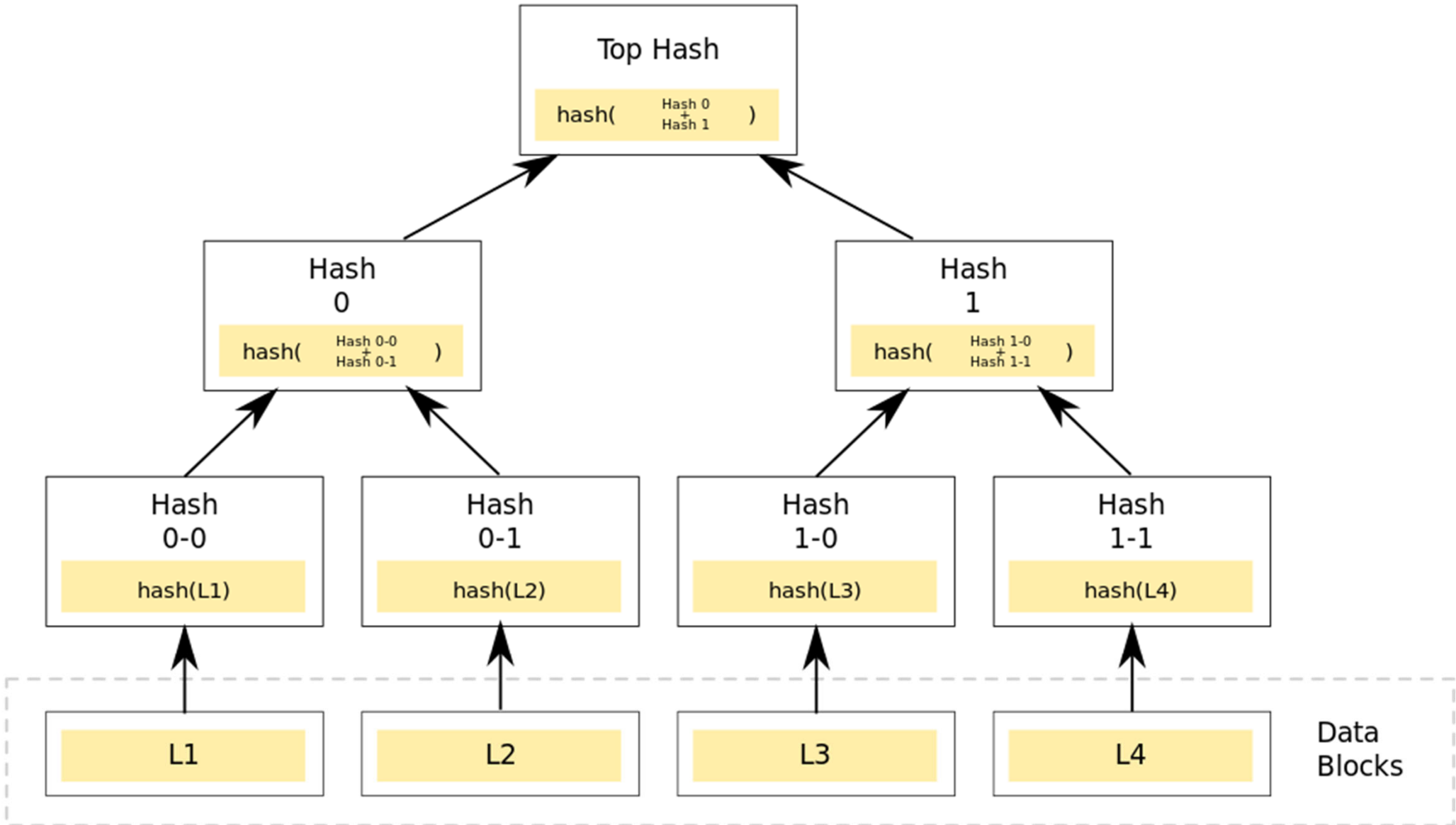
- **Fully private blockchains:** write permissions are kept centralized to one organization. Read permissions may be public or restricted to an arbitrary extent. Likely applications include database management, auditing, etc internal to a single company, and so public readability may not be necessary in many cases at all, though in other cases public auditability is desired.

# Outline

- Introduction
- Blockchain properties
- Bitcoin transactions
- Consensus
- Types of blockchains
- **Merkle trees**
- Smart contracts
- dApps

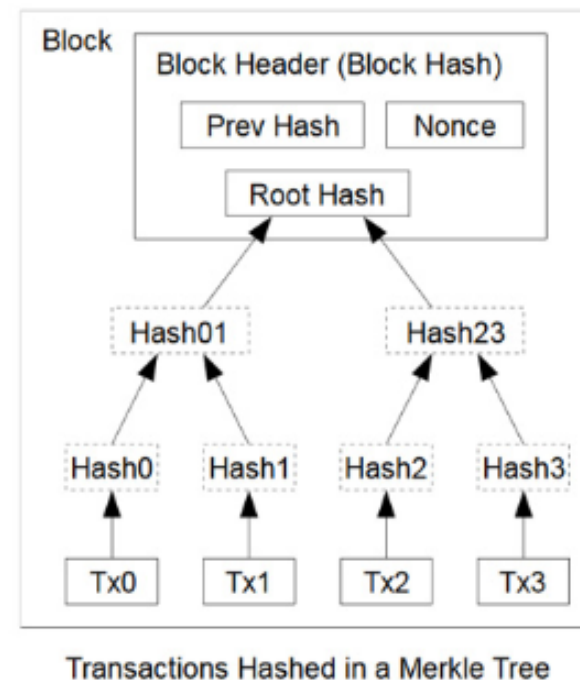
# Merkle trees

- a tree in which every leaf node is labelled with the hash of a data block, and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.
- Demonstrating that a leaf node is a part of a given binary hash tree requires computing a number of hashes proportional to the logarithm of the number of leaf nodes of the tree
- As long as the root hash is publicly known and trusted, it is possible for anyone who wants to do a key-value lookup on a database to use a Merkle proof to verify the position and integrity of a piece of data within a database that has a particular root



# Merkle Trees in Bitcoin

- blocks in a blockchain are connected through hashes of the previous block
- In Bitcoin, each block contains all of the transactions within that block as well as the block header which consists of:
  - Block Version Number
  - Previous Block Hash
  - Timestamp
  - Mining Difficulty Target
  - Nonce
  - Merkle Root Hash



# Outline

- Introduction
- Blockchain properties
- Bitcoin transactions
- Consensus
- Types of blockchains
- Merkle trees
- **Smart contracts**
- dApps

# Smart contracts

- Computer protocols that can verify, facilitate or enforce negotiation or performance of a contract
- Eliminate middlemen/intermediaries
- Asset -> algorithm -> asset destination
- Domains : government (voting), management (automatic workflows), supply chain, real estate, healthcare ( ensure confidentiality, store immutable data)
- Smart contracts – programs that are run on the blockchain (e.g. Ethereum – blockchain + Turing complete language)

# Principles of contract design

- Observability – principals can observe each others' performance of the contract, or to prove their performance to other principals (accounting)
- Verifiability - principal can prove to an arbitrator that a contract has been performed or breached, or the ability of the arbitrator to find this out by other means (auditing)
- Privity - knowledge and control over the contents and performance of a contract should be distributed among parties only as much as is necessary for the performance of that contract (security)
- Enforceability – contracts can be enforced



# Smart contract features

- Deterministic – always same output to given input.  
Effects: avoid calling un-deterministic system functions, data resources or dynamic calls
- Terminable. Solutions : Turing incompleteness, agree beforehand on payment for max execution steps, timer.
- Isolated. Smart contracts are run in virtual machines or containers

# Contract example (Solidity language)

```
pragma solidity >=0.4.22 <0.6.0;

contract SimpleAuction {
    address payable public beneficiary;
    uint public auctionEndTime;
    // Current state of the auction.
    address public highestBidder;
    uint public highestBid;
    // Allowed withdrawals of previous bids
    mapping(address => uint) pendingReturns;
    bool ended;
    event HighestBidIncreased(address bidder, uint amount);
    event AuctionEnded(address winner, uint amount);
    constructor(
        uint _biddingTime,
        address payable _beneficiary
    ) public {
        beneficiary = _beneficiary;
        auctionEndTime = now + _biddingTime;
    }
}
```

```
function bid() public payable {
    require(
        now <= auctionEndTime,
        "Auction already ended."
    );
    require(
        msg.value > highestBid,
        "There already is a higher bid."
    );
    if (highestBid != 0) {
        pendingReturns[highestBidder] += highestBid;
    }
    highestBidder = msg.sender;
    highestBid = msg.value;
    //Allow recipients to withdraw the money themselves
    emit HighestBidIncreased(msg.sender, msg.value);
}
```

```
/// Withdraw a bid that was overbid.  
function withdraw() public returns (bool) {  
    uint amount = pendingReturns[msg.sender];  
    if (amount > 0) {  
        pendingReturns[msg.sender] = 0;  
        if (!msg.sender.send(amount)) {  
            // No need to call throw here, just reset the amount owing  
            pendingReturns[msg.sender] = amount;  
            return false;  
        }  
    }  
    return true;  
}  
function auctionEnd() public {  
  
    require(now >= auctionEndTime, "Auction not yet ended.");  
    require(!ended, "auctionEnd has already been called.");  
  
    ended = true;  
    emit AuctionEnded(highestBidder, highestBid);  
  
    beneficiary.transfer(highestBid);  
}  
}
```

# Cost of transactions on Ethereum

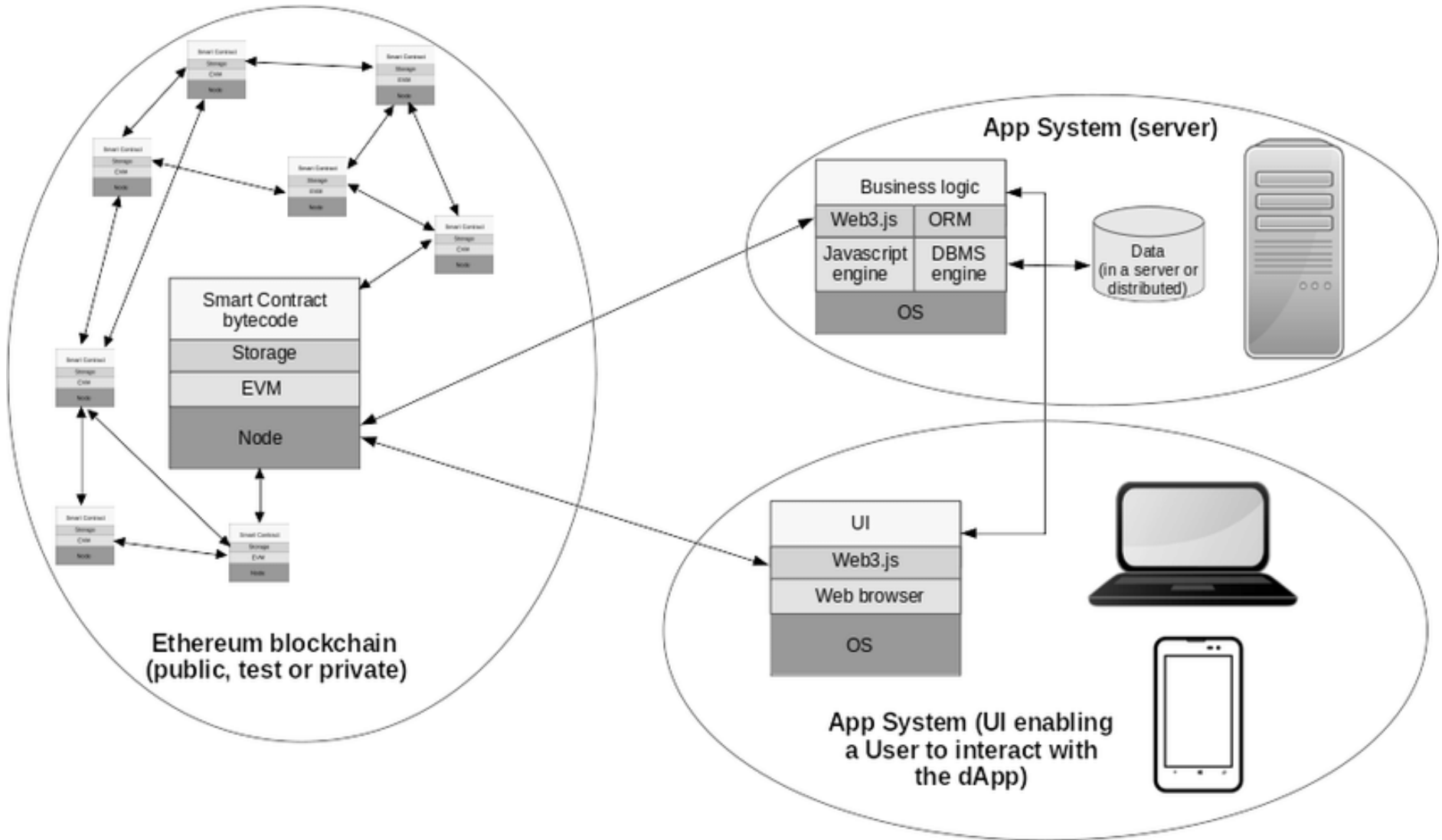
- Operations (transactions or smart contract executions) require gas
- Gas measures computational effort for operation execution
- Miners that execute operations are paid in Ether corresponding to the gas required
- Gas price (in Ether) is specified by the sender of transaction so miners can choose transactions with highest prices
- Gas limit – maximum gas sender is willing to pay for a transaction

# Outline

- Introduction
- Blockchain properties
- Bitcoin transactions
- Consensus
- Types of blockchains
- Merkle trees
- Smart contracts
- **dApps**

# Decentralized Applications (dApps)

- Applications running on a peer to peer network of computers (such as blockchain). We`ll refer to dApps running on blockchain.
- Core logic of a dApp are smart contracts interacting with the blockchain. Frontend can be the same to traditional applications.
- Frontend contains a “wallet” (managing the digital identity, private and public keys) communicating to the smart contract by sending triggers and receiving events.
- E.g. WeiFund – crowdfunding platform, similar frontend to traditional crowdfunding platforms but backed by smart contracts





# Bibliography

- <https://bitcoin.org/bitcoin.pdf>
- <https://www.pcmag.com/news/blockchain-the-invisible-technology-thats-changing-the-world>
- <https://bitcoin.org/en/glossary/unspent-transaction-output>
- <https://onezero.medium.com/how-does-the-blockchain-work-98c8cd01d2ae>
- <https://cointelegraph.com/bitcoin-for-beginners/what-are-cryptocurrencies#accept-as-payment-for-business>
- <https://hbr.org/2017/03/how-safe-are-blockchains-it-depends>
- [https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart\\_contracts\\_2.html](https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html)
- <https://solidity.readthedocs.io/en/v0.4.24/solidity-by-example.html>