Blockchain Essentials

Pahontu Bogdan-Ionut
E-mail: pahontubogdan@gmail.com

Agenda

- What is Blockchain ?
- Blockchain Types
- Blockchain Architecture / Transactions
- Consensus Mechanisms
- Blockchain Evolution
- Smart Contracts
- ERC
- Blockchain DApps
- Use cases
- Notable Tokens
- Questions

What is Blockchain ?

- Blockchain → Data structure consisting of packages / blocks that are connected forming a digital chain
- A distributed database with or without a controlling authority
- **Distributed Ledger Technology** (DLT) → records and shares data across a distributed network
- Mechanism for storing cryptocurrencies
- Trusted architecture
- Before adding it to the chain data is validated by miners/validators.

Basic Terms

- **Blockchain:**
- **Validators or miners**
- **Consensus mechanism**
- **Cryptocurrencies**
- **SmartContracts**
- **Distributed Ledger**

Important Fact:

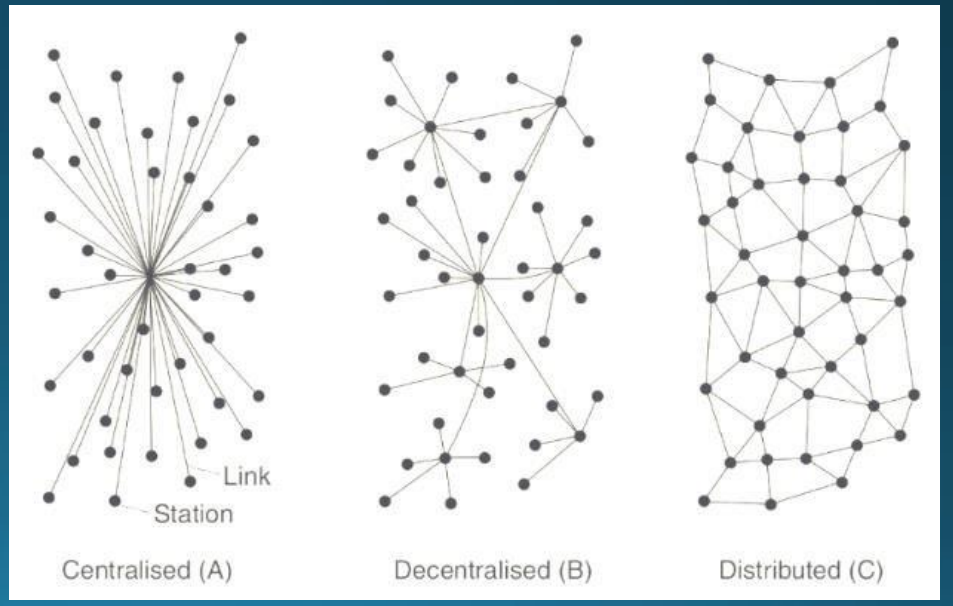
Cryptocurrencies != Blockchain

Characteristics

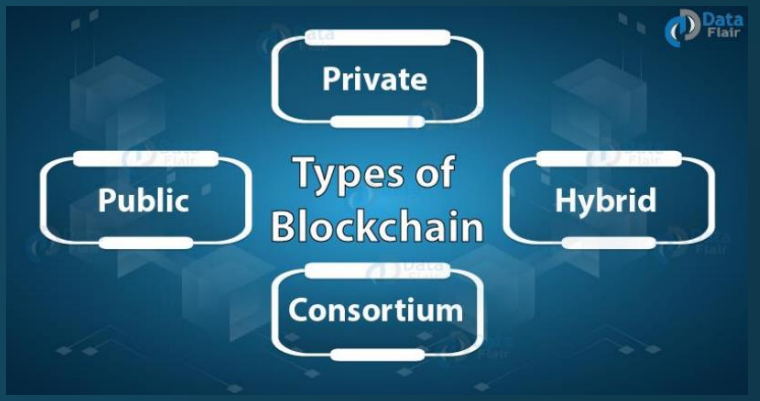
- **Decentralization**
  - **Privacy**
  - **Reliability**
  - **Versatility**
- **Trust**
  - **Transparency**
  - **Data Integrity**
  - **Data Immutability**

Distributed Ledger (DLT)

- Distributed, replicated "database"
- Distribution via a p2p model (no master)
- Consensus model used to ensure integrity
- Append only "immutable" store
- Highly fault tolerant
- Eventually consistent

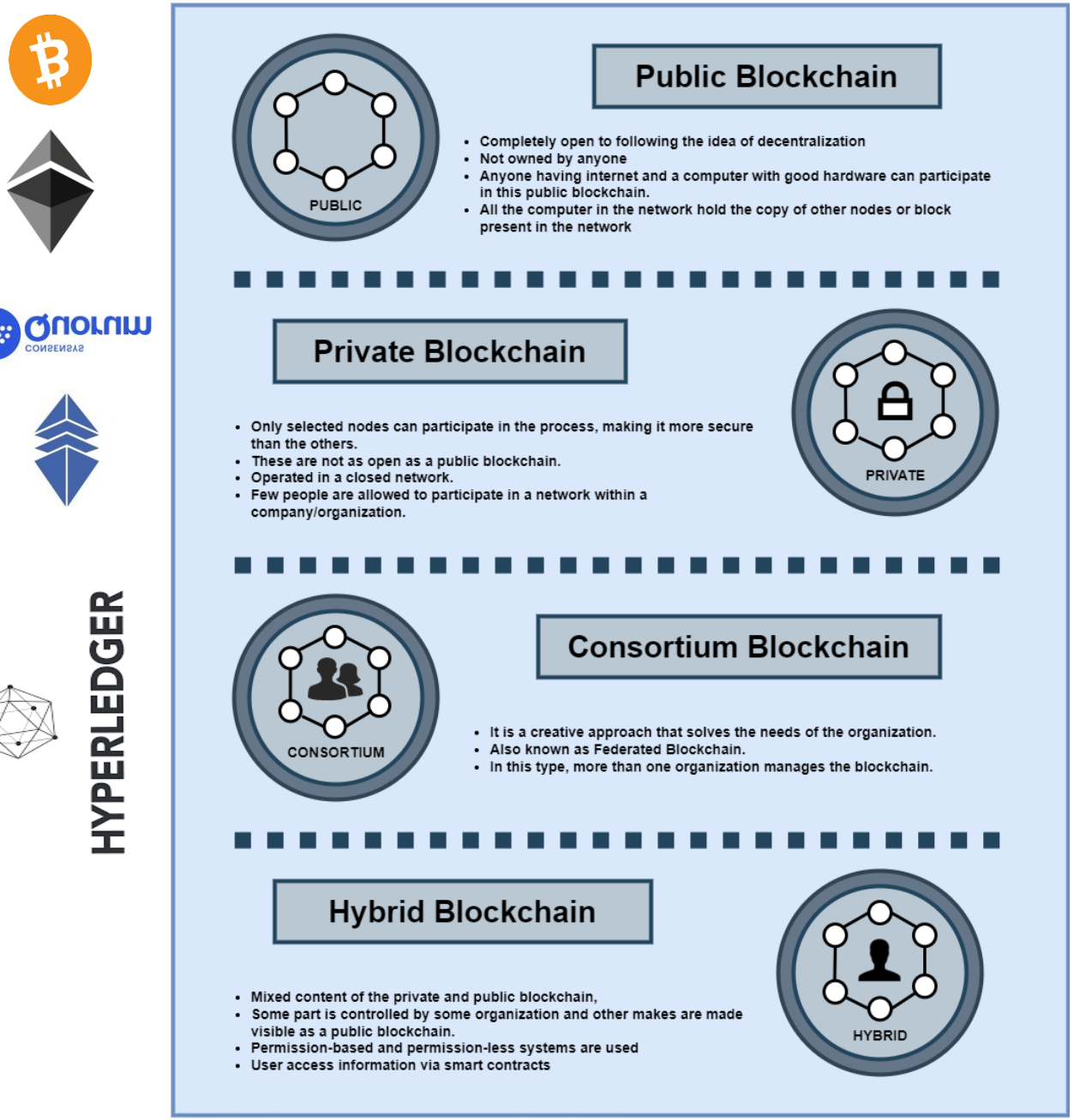


Blockchain Types (I)



- Public blockchains like Bitcoin and Ethereum
  - ➢ Non-restrictive, permission-less distributed ledger system
- Private blockchains like Hyperledger and R3 Corda
  - ➢ Restrictive or permission blockchain operative only in a closed network
- Consortium blockchains like Energy Web Foundation, R3
  - ➢ Semi-decentralized type where more than one organization manages a blockchain network
- Hybrid blockchains like Dragonchain
  - ➢ Combination of the private and public blockchain

Blockchain Types (II)

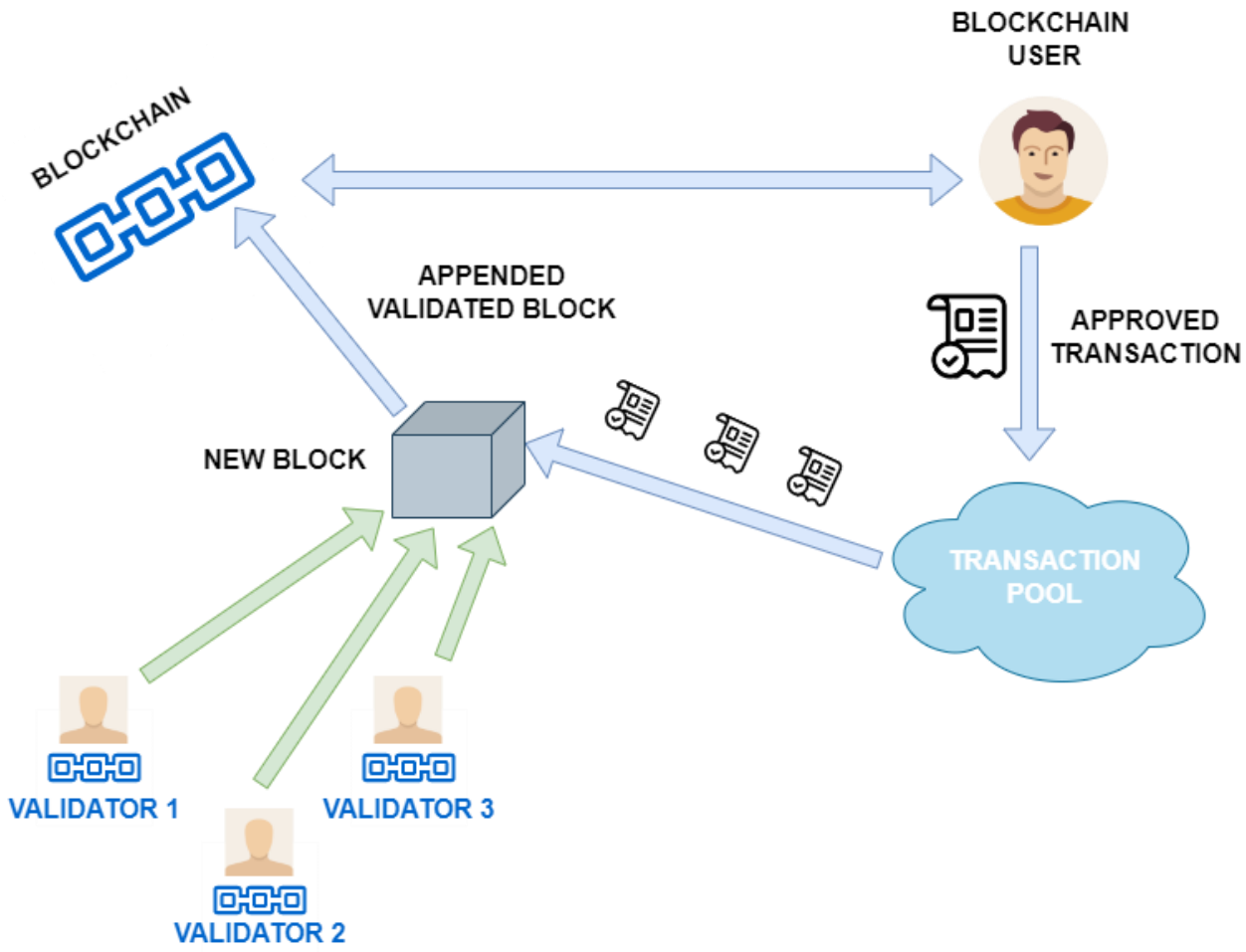


HYPERLEDGER

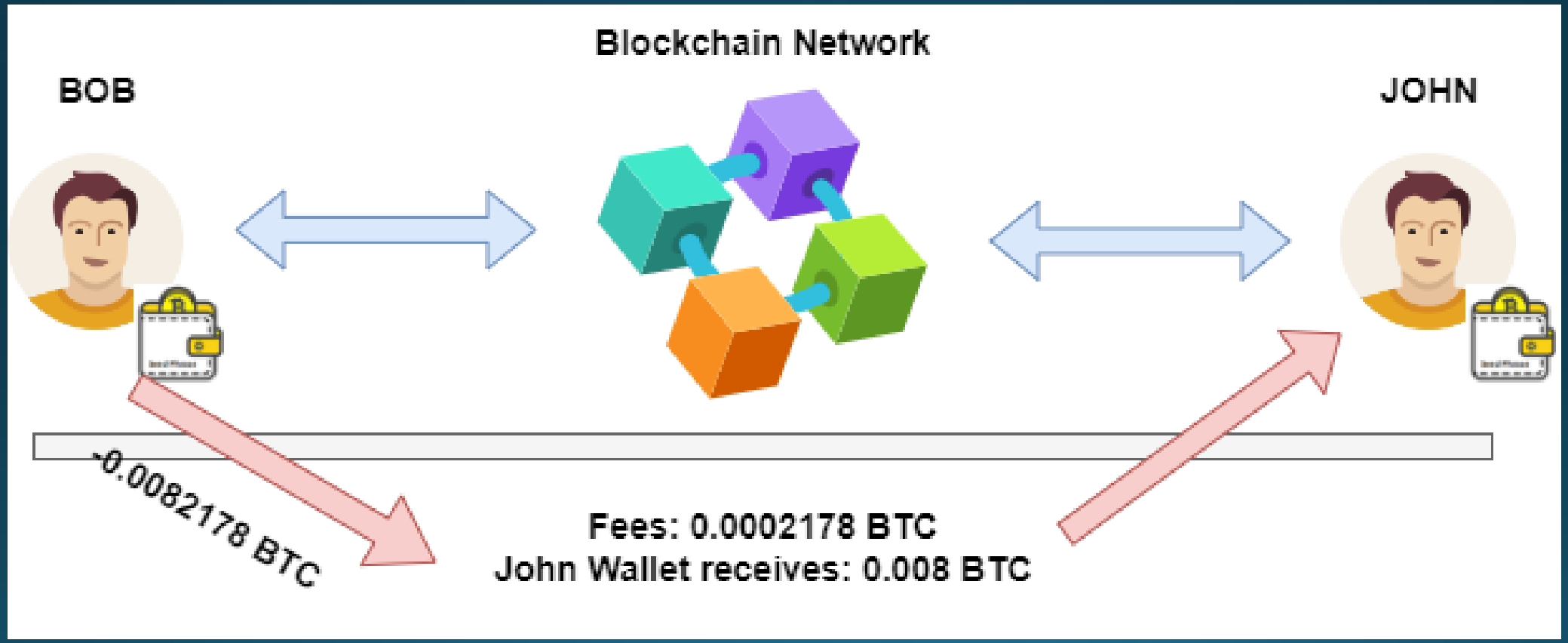
MultiChain

DRAGONCHAIN

Simple Architecture

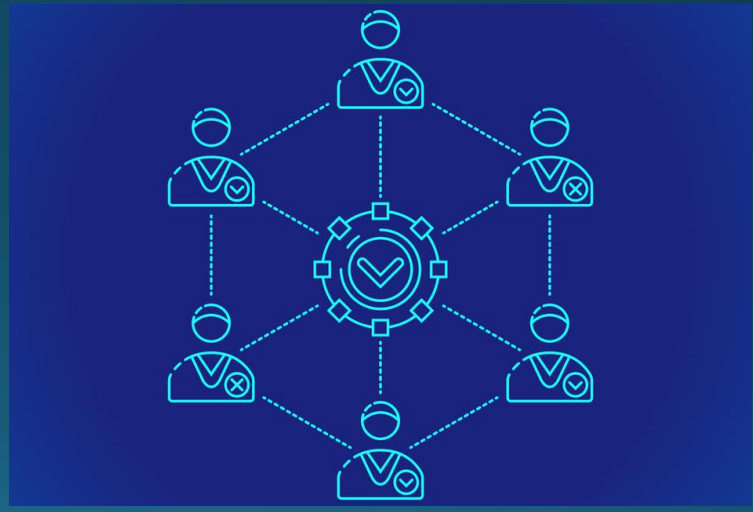


Blockchain Transaction



Consensus (I)

- Q. How do we know that the information on chain is correct ?
- A. Because a “consensus” mechanism is used and everyone agrees – this is “consensus”
- What does this mean ?
  - ➢ Participatory
  - ➢ Egalitarian
  - ➢ Agreement Seeking
  - ➢ Collaborative
  - ➢ Cooperative
  - ➢ Inclusive

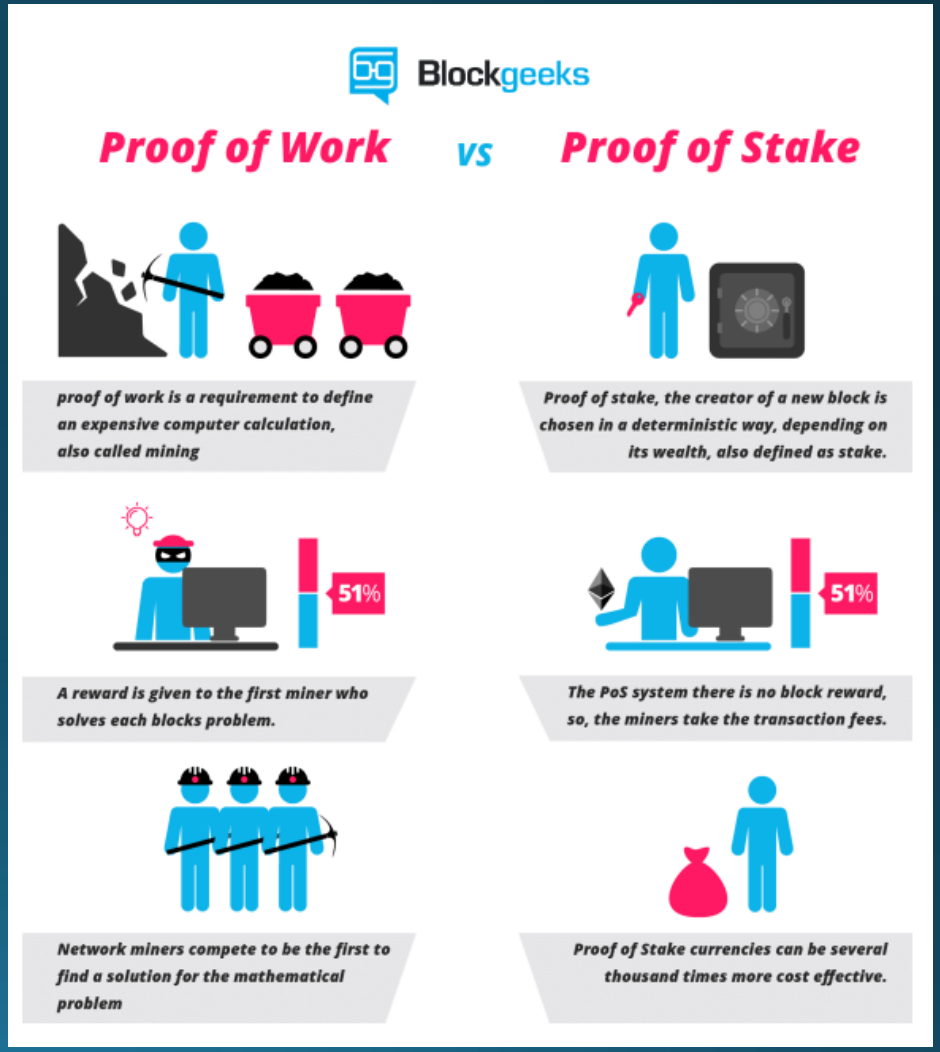


Consensus (II)

- **Proof of Work**
- **Proof of Stake**
- **Delegate Proof of Stake**
- **Hybrid Consensus**
- **Practical Byzantine Fault Tolerance**

PoW vs PoS

- Proof of Work (PoW)
  - To create (“mine”) a block you need to solve a hard problem
  - If you don’t solve the problem, then peers will reject your block
  - Forging the blockchain would require a huge amount of work to get your fraudulent blocks accepted
  - Examples: Bitcoin, Ethereum (1.x), Litecoin, etc.
- Proof of Stake (PoS)
  - A person can validate block transactions according to how many coins he or she holds
  - Allocation of responsibility based on the amount “staked”
  - Does not require high amount of resourced used as in PoW
  - Examples: Ethereum (2.0), Polygon, etc.

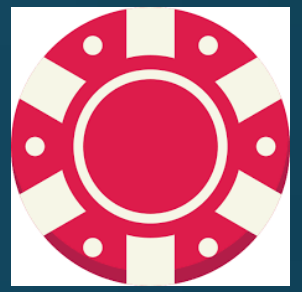


Blockchain Generations

- First Generation: (ex: Bitcoin)
  - The reason why the first-gen blockchain – specifically Bitcoin – was created was to radically improve the monetary systems in place
  - Bitcoin is the first real use case of blockchain technology and its main purpose is as a financial application
  - Bob can send Bill digital money and there is security in that transaction
  - Both can enjoy privacy because the transaction is anonymous
- Second Generation: (ex: Ethereum)
  - Smart contracts
  - Acts as a platform which developers can use to build on, like apps have iOS, decentralized apps (dApps) have Ethereum
  - Smart Contracts, dApps, launch platforms for ICOs like Ethereum, Polygon, Elrond
  - Variety of functional uses including decentralized finance (DeFi), web browsing, gaming, identity management, supply chain management
- Third Generation: (ex: ETH 2.0, Polkadot)
  - Too many people trying to transact and too little space for it on the blockchain
  - Scalability
  - Interoperability
- Next Generation: (ex: Deep Brain Chain)
  - 3rd generation blockchains that have implemented AI

Coins vs Token

- Used to pay network fees
- Usually has its own blockchain
- Native asset of the protocol, not described by a standard
- Used for value transfer between parties

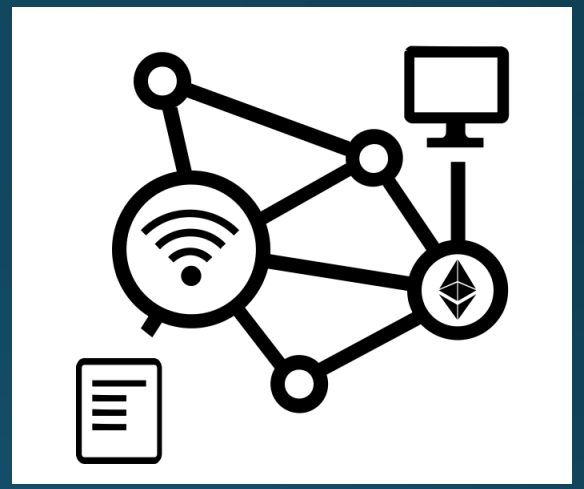


- Created usually by a project
- Does not have its own blockchain
- May implement a standard (like ERC-20, ERC-721, ERC-1155)
- Interacts with blockchain via smart contracts, used in dApps

Smart Contracts (I)

A contract between parties is written as code into the blockchain.

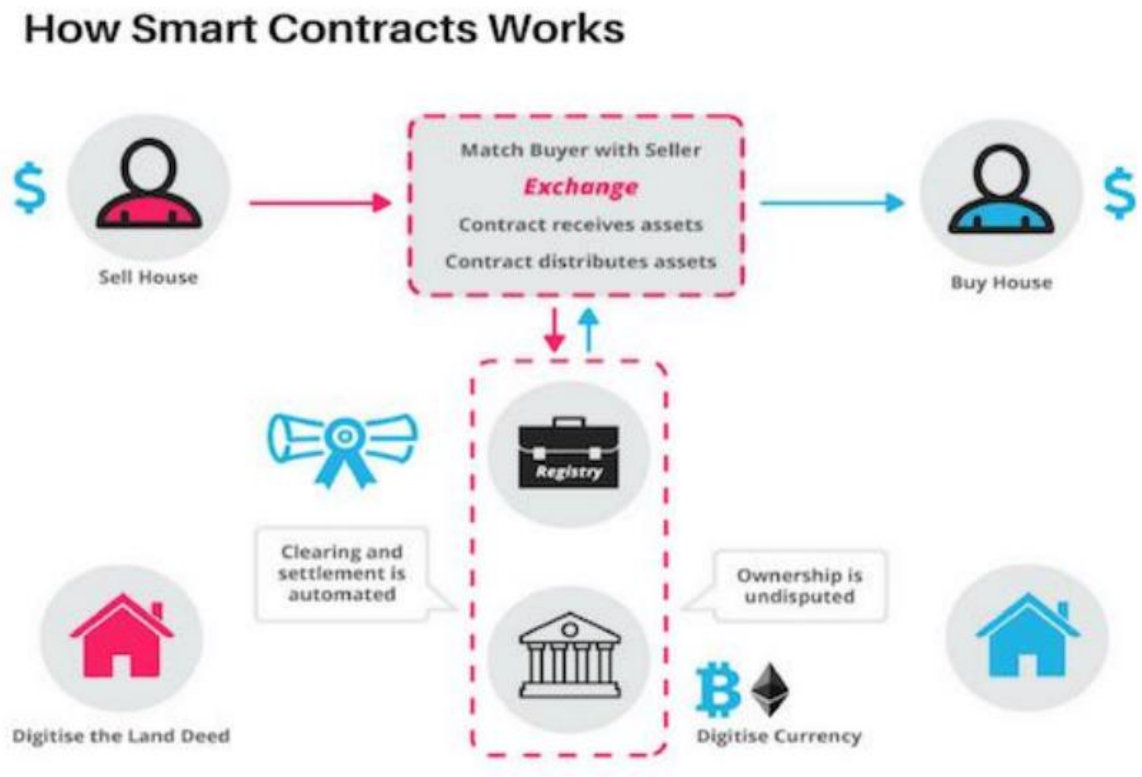
The individuals involved are anonymous, but the contract is in the public ledger.



A triggering event like a new credit request is hit and the contract executes itself according to the coded terms.

Smart Contracts (II)

- Code stored in a blockchain, executed by the blockchain virtual machine
  - ➢ Executed when certain conditions are met
  - ➢ It is publicly available
  - ➢ It resembles a notary act
- Programming models vary by platform
  - ➢ Bitcoin - primitive ”Forth” script
  - ➢ IBM Hyperledger – GoLang and JavaScript
  - ➢ Ethereum - Solidity - one of the most popular example



Smart Contracts (III)

- Autonomy – There's no need to rely on a broker, lawyer or other intermediaries to confirm.
- Trust – Documents are encrypted on a shared ledger – it can't be lost
- Backup – Documents are duplicated many times over on the blockchain
- Safety – Cryptography, the encryption of websites, keeps your documents safe
- Speed – Use software code to automate tasks, thereby shaving hours off a range of business processes
- Savings – Save money since they knock out the presence of an intermediary
- Accuracy – Avoid errors that come from manually filling out heaps of forms

ERC Tokens

What is ERC ?

ERC-20

✓ Smart contract implementation for tokens

✓ Rules for creating a token

✓ Tokens can be transferred

✓ Tokens are fungible (tokens of same type are identical)

ERC-721

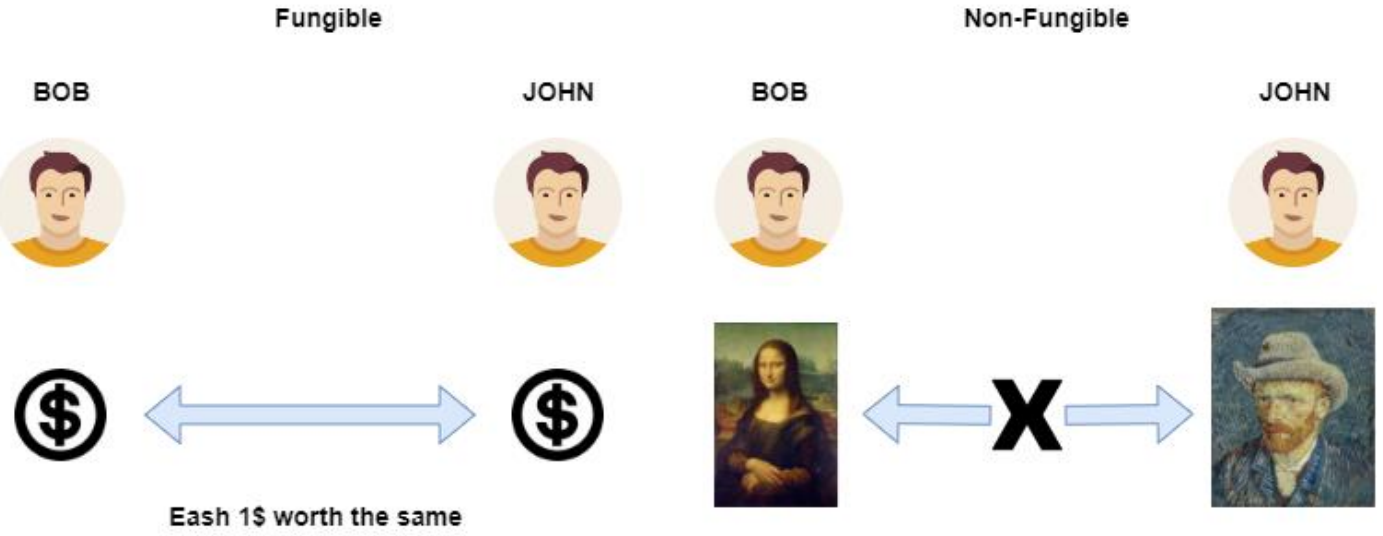
✓ Each token can be unique

✓ Tokens can be different (age, rarity, visual, etc.)

✓ Tokens can be transferred

✓ Tokens are non-fungible

Fungible vs Non-Fungible Tokens



Non-Fungible Tokens Properties

- Unique;
- Easy Transferable;
- Guaranteed ownership;
- Fraud Proof;
- Indivisible;
- Provable Scarce;

ERC Tokens (II)

ERC-1155

✓ Each token has multiple instances of same base

✓ Tokens can be different (age, rarity, visual, etc.)

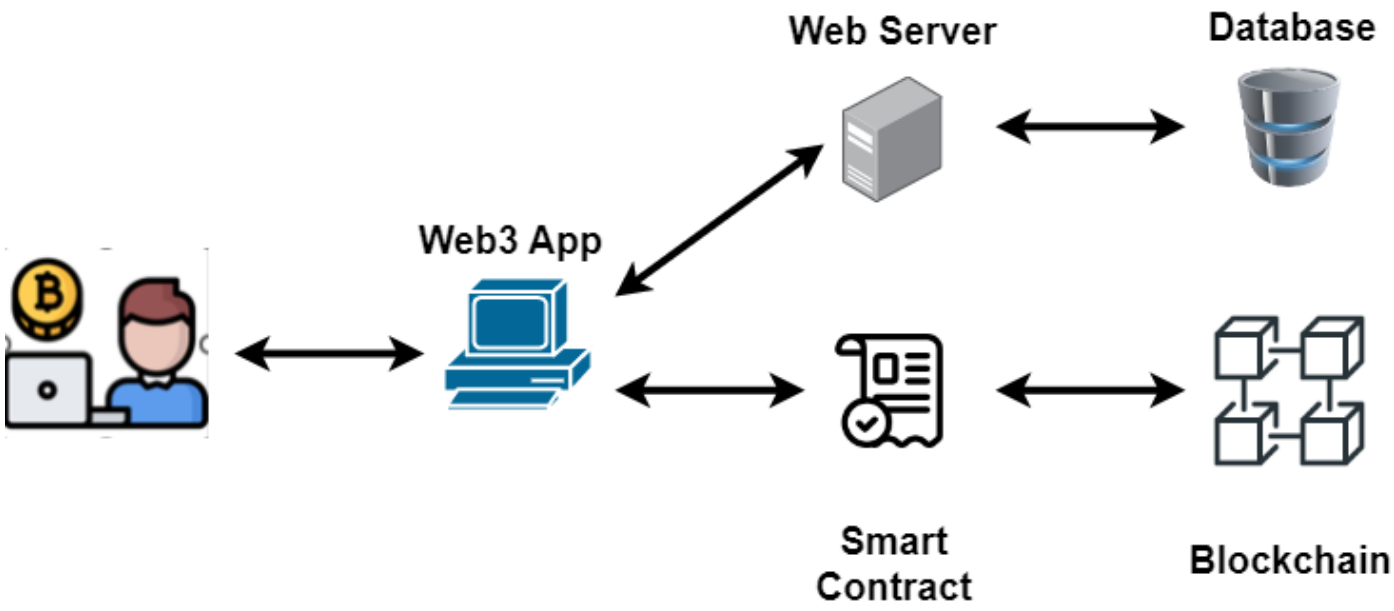
✓ Tokens can be transferred

✓ Tokens are non-fungible

Example: Shields in games

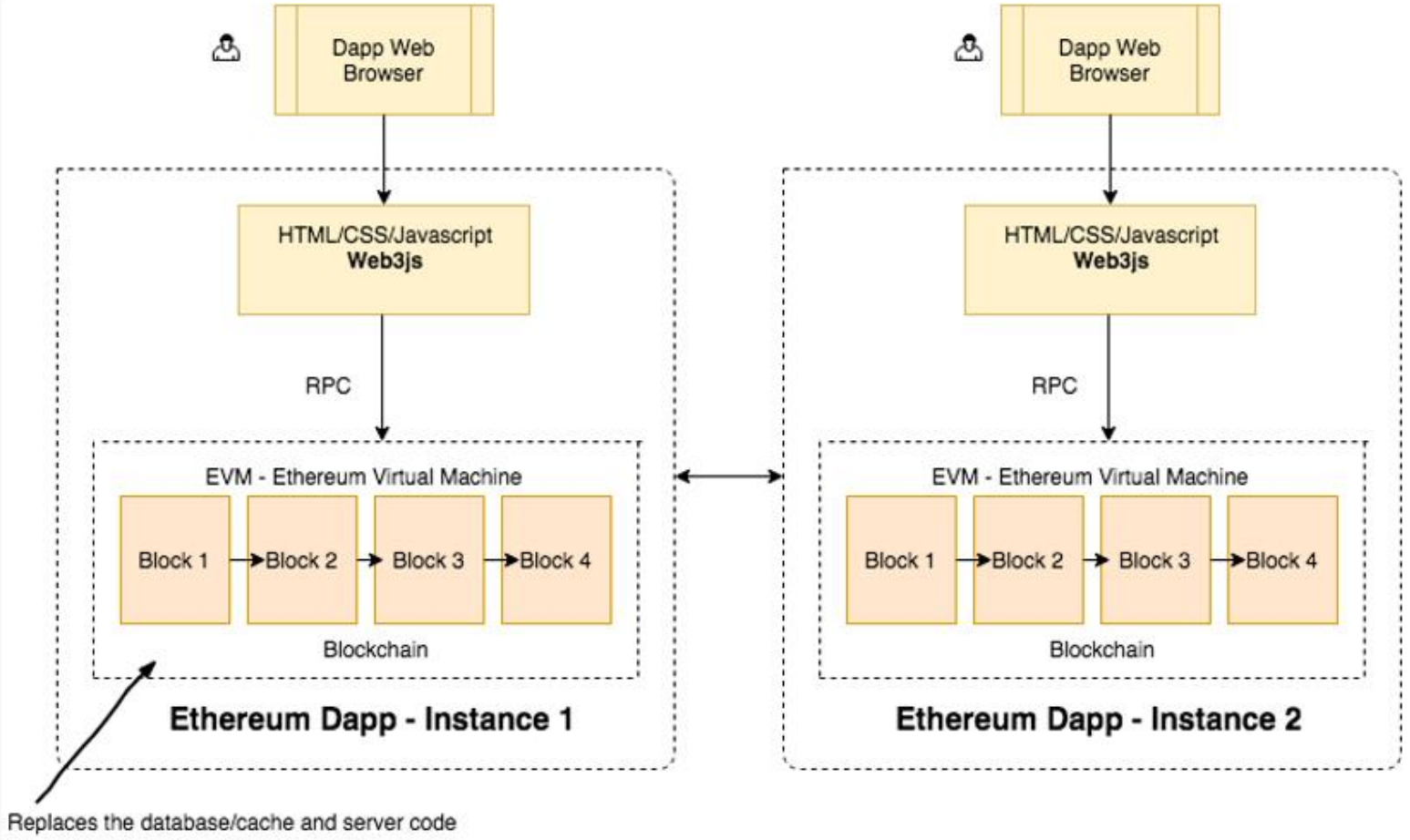
Blockchain Apps

Digital Apps (Swapping, Borrowing, Staking, etc..)

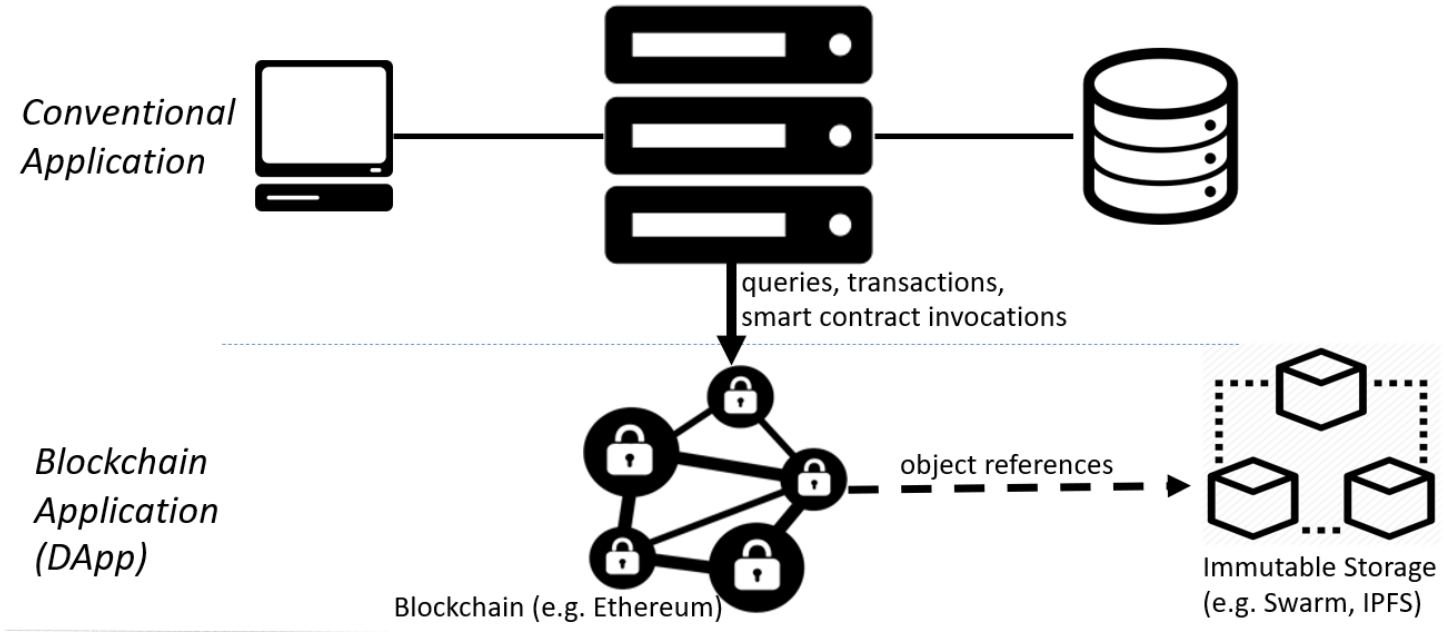


dApp Architecture

- View layer which controls what the user sees
- Controller layer which synthesizes which actions need to be performed
- Blockchain (Smart Contracts) layer which manages the data, logic and rules of the application in a distributed ledger



Blockchain Integration



Blockchain Use Cases

- Cryptocurrencies;
- Transfer Values;
- Identity;
- Storage;
- Real Estate;
- Logistics

NFT Use Cases

- Gaming:
  - Decentraland;
  - Axie Infinity;
- Domain Names:
  - Unstoppable Domains;
- Digital Art:
  - OpenSea;

Notable blockchains

Other Notable tokens

Questions ?