

### **What would you have gained by the end of this course?**

- Working knowledge of several tools and techniques at the intersection of blockchain and AI
- Perform smart contract dev on several blockchains
- Deploy and run decentralized machine learning code on federated learning frameworks
- Critically assess the research in AI and blockchain
- Develop a project that leverages techniques in AI and Blockchain

### **What is blockchain?**

- The Record
  - Can contain any form of data/information.
- The Block
  - A bundle of records
- The chain
  - All the blocks linked together

### **Steps:**

1. Transaction
  - a. Transferring cryptocurrency and information involving who sent it, who will receive it, amount, etc. Kind of a “duh” definition but whatever.
2. Distributed consensus
  - a. Is the thing the real thing, prevents counterfeiting.
  - b. Blockchains are decentralized and maintained by multiple hosts/computers. Everyone has a shared view of the blockchain’s state.
3. Block Creation
  - a. Each block has a hash and is stored in a larger network of blocks.
  - b. Collection of transactions and information that contains a unique hash and also the hash of the previous block, like a linked list.
4. Adding the block to the blockchain

- a. The block is validated and agreed via the network of nodes/computers. If there's a collective agreement, or consensus, then the block will be added to the blockchain.

*Note: A transaction in a committed block is difficult to change*

### **Blockchain != Cryptocurrency**

- Cryptocurrency is an application that sits on top of a blockchain

### **Blockchain Pillars:**

- Authenticity (cryptographic): creates transactions that are impervious to fraud through the use of public/private signatures establishing a shared truth
- Shared: The more companies participating in the blockchain the more value it brings
- Distributed: many replicas of the blockchain database making it more authentic
- Ledger: read/write once database maintaining an immutable record of every transaction
  - It's immutable, because if you were to change it, it would mean changing the hash of the block.

### **The disruptive effect in Blockchain**

- Flatten ecosystem and supply chain removing middleman processes
- Peer-to-peer value exchange reducing settlement time
- One ledger instead of comparing multiple ledgers
- More collaborative economy - shared costs, risks, etc
- Dramatic changes in how identity is defined and controlled

### **What is AI?**

- Simulation of intelligence properties in machines
- The science of making intelligent machines through intelligent software

- The study and making of machines that mimic human thinking and goes beyond what humans are capable of doing
- Ability to use massive computing power to address certain situations, predict what might happen, and proactively do something about it

## **AI and Blockchain**


- Blockchain from AI perspective:
  - Open data for ML
  - AI data exchange in trustless environments
  - AI evolution towards human AI
- AI from blockchain perspective:
  - Economical agents
  - Data consumers
  - Data producers
  - Minus (proof of cognitive work)

## **Blockchain technology has:**

- Created an industry worth hundreds of billions of dollars
- Launched a wave of innovation in distributed systems, cryptography, privacy, security and economics

## **Two views:**

- Some believe that blockchains will be integral to the future of money, governments and the internet
- Others claim that this is a transient bubble

 [The Most Elusive Identity On The Internet - Pt. 1 \(Ft. Nexpo\)](#)

## **Bitcoin's inspirations:**

- Ralph Markle's work on Merkle trees
- Haber and Stornetta's work on cryptographic timestamping services
- Hashcash by Adam Back
- b-Money by Wei Dai

## **Bitcoin's Primary Innovation:**

- Proof of work Consensus (now called the Nakamoto consensus)

## **The cypherpunks:**

*Cypherpunks write code. We know that someone has to write software to defend privacy, and we're going to write it*

- Eric Hughes, the Cypherpunk Manifesto

## **Cypherpunk Economics**

- The cypherpunks were deeply suspicious of central banks and their control over monetary policy
  - Taxation
  - Seigniorage (virtually printing money)
- Taxation generally requires the assent of citizens, whereas printing money can be done unilaterally. Cypherpunks saw money printing as a form of theft from the currency holders

## **The Double Send Problem**

If money is just bits on a computer, why can't they just copy-paste the data?

Paypal solves this by having a central database with user accounts

<https://nakamoto.com/>

ecash/digicash - early (1986) predecessor to bitcoin

## **B-money and BitGold:**

- Public key cryptography for identity
- Proof-of-work to mint new coins (bitcoin uses proof-of-work to update the blockchain and append transactions)
- Trusted timestamping servers for transaction ordering
- Achieves consensus by counting the total nodes in the network and letting the nodes vote (bitcoin achieves consensus by counting the total work performed in the network)

- Vulnerable to Sybil Attacks.

## **Cryptographic Identities**

- Private keys/Public Keys
  - Private key is just a random number.
  - The public key is mathematically linked to the private key.
  - It is easy to go from the private key to the public key—but very difficult to go from public to private.
- Public address
  - This is either identical to the public key (E.g., Ethereum) or a function of the public key (e.g. Bitcoin)
    - public key → cryptographic hash function → address
- If a user loses a private key, then any digital assets associated with it are lost.
- If stolen, the attacker will have full access to all digital assets controlled by that private key.

## **Consensus Mechanisms**

- Consensus is an agreement among a group of people on an idea, statement, or plan of action.
  - Majority: 51%
  - Supermajority: 66% (sometimes higher)
  - Unanimous: 100%
  - Weighted: Not all votes weighed equally or multiple votes per agent
- Consensus is typically only relevant when there is no distinguished leader
  - A jury must reach a consensus on a court verdict (unanimous)
  - The senate must reach a consensus on new bills being passed (majority or supermajority)
- Particularly important when there is significant disagreement or potential for untrustworthy parties in the discussions around the decision.

## Need for Consensus

- Consensus is a very difficult problem when parties are not trusted
- The network must maintain integrity in order to maintain value
- Past transactions must be trusted for the network to function
- Thus, the ability to verify transactions without trust is needed
- The consensus problem can often be rephrased as the ability to trust the result of a transaction or block without trusting the parties involved in the transaction, or the party that verified it.

## Byzantine Fault Tolerance:

- A system is considered to be Byzantine Fault Tolerant (BFT) if it is resistant to the dilemmas of the Byzantine Generals Problem
- A Byzantine Fault is defined as a failure mode of the system, either failing to function or functioning incorrectly, caused by an inability to reach or error in consensus among the system
- Most consensus algorithms used in blockchain technologies are Byzantine Fault Tolerant

You don't have to pay the person immediately when you make a trade, you can give them the **promise** that you'll pay them back later. An example of this is a credit card. This can have the downside that you expose your credit card number on said platform

## How to send messages securely on the Internet:

Use a public and private key

- You have a private key and a public key derived from the private key
- The message is signed and checked against the other's key by a known system

Example: given a public modulus by some  $n$ , users sign messages by  $(m * s) \bmod n$

.....finish this later

## Blockchain Trilemma

Triangle of scalability, security, decentralization

**Hard fork:** blocks that have not been updated will not work with changed blocks as the new forks will be rejected

**Soft fork:** backwards compatible with blocks that have not been updated

	Permissionless Blockchain	Permissioned Blockchain	Database
Throughput	Low	High	Very High
Latency	Slow	Medium	Fast
Number of readers	High	High	High
Number of writers	High	Low	High
Number of untrusted writers	High	Low	0
Consensus mechanism	PoW, PoS, etc.	BFT Protocols	None
Centrally managed?	No	Yes	Yes

### Quantum-proof crypto?

Not all current cryptos are resilient in the face of advances in quantum computing

- Algorithms that rely on the **computational complexity of integer factorization** (such as RSA) or work on **solving discrete logarithms** are very susceptible to being broken by quantum computing
- The hashing algorithms used by blockchain networks are much less likely to be cracked, as you can just increase the hash size or output

### Weak Immutability

- If two chains are competing but each includes its own unique sequence of tail blocks, whichever blockchain is longer will be adopted
- Transactions within the replaced blocks might be included in a different block or added back to the pending transaction pool
- Depending on the size of the blockchain network, this could be a very cost-prohibitive attack carried out by state-level hackers

- In blockchains (in general) there is no such thing as perfect immutability
- There are some examples where mutability is welcomed (like in private blockchains)

## The Oracle Problem

Blockchains might have trouble interacting with the real world because on a blockchain events are recorded in a specific order, unlike IRL where the order is not kept track of

Some solutions to this problem include:

- Centralized: Oraclize, Chainlink, etc
- Mineable oracle contracts

**Situations where oracles in blockchains don't work in blockchain systems:**

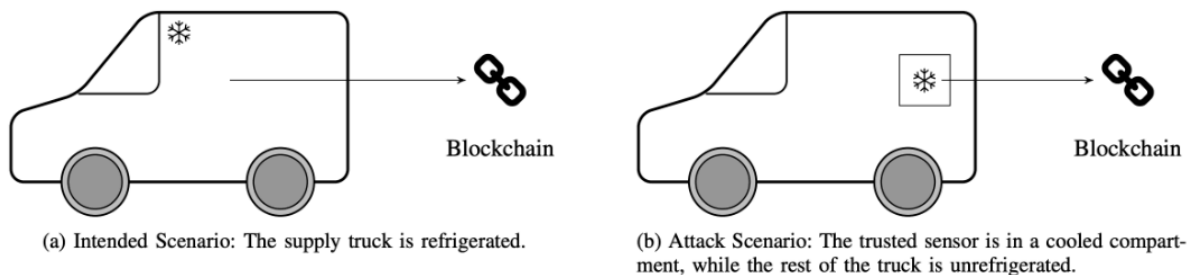


Fig. 3: An example for an attack that can be conducted against a tamperproof temperature sensor that writes collected data to the blockchain to ensure proper cooling of goods in a supply chain. The left subfigure shows the intended situation, where the delivery truck containing the goods is refrigerated, the trusted sensor measures the temperature and publishes the data to the blockchain correctly. The right subfigure shows the possible attack, where the supplier – e.g. to save costs – only cools down a small fridge inside of the truck in which he puts the sensor, while the goods are in the non-refrigerated section of the truck.

## Users involved in blockchain governance

- Blockchain networks have some level of control/ownership that depends on the type of system the blockchain is running on.
  - If *permissioned*, they are generally set up by a owner/consortium, which governs the network



- If *permissionless*, they are governed by blockchain network users, publishing nodes and software developers (so pretty much all decentralized entities).
- Most blockchains are open source and it is possible to inspect the source code and compile it independently. Not every user can do this though, it's mostly likely a restriction that can only be sorted by the person in charge (in permissioned.)

### **Cyber networks and vulnerabilities:**

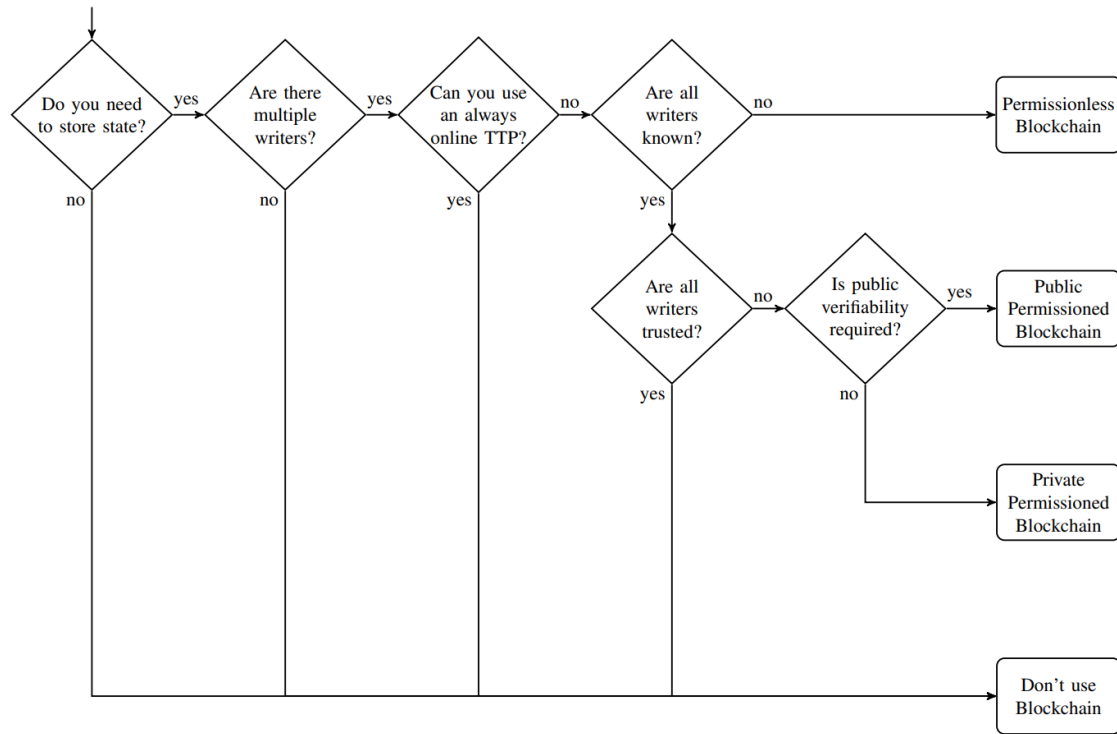
- Transactions that have not yet been included in a published block within the blockchain are vulnerable to several types of attacks
  - Spoofing time in transactional timestamps
  - Adjusting the clock of a member of an ordering service
- Malicious users
  - Network scanning and reconnaissance to discover and exploit vulnerabilities and launch zero-day attacks
  - Ignoring transactions from specific users, nodes, or even entire countries
  - Creating an altered, alternative chain in secret, then submitting it once the alternative chain is longer than the real chain
  - Refusing to transmit blocks to other nodes, essentially disrupting the distribution of information (this is not an issue if the blockchain network is sufficiently decentralized)
- Smart contract bugs

### **Legislation:**

- Crypto bans
  - Egypt, Iraq, Qatar, Oman, Morocco, Algeria, Tunisia, Bangladesh, and China have all banned cryptocurrency
  - Forty-two other countries have implicitly banned digital currencies by putting restrictions on the ability of banks to deal with crypto or prohibiting cryptocurrency exchanges

- Climate-change and energy-specific legislation

## Do You Need a Blockchain?



### Application Consideration:

Blockchain technology solutions may be suitable if the activities or systems require features such as:

- Many distributed participants
- Workflow is transactional in nature (e.g., transfer of digital assets/information between parties)
- A need for:
  - lack of trusted third party
  - globally scarce digital identifier (i.e., digital art, digital land, digital property)
  - decentralized naming service or ordered registry
  - a cryptographically secure system of ownership

- reducing or eliminating manual efforts of reconciliation and dispute resolutions
- enabling real-time monitoring of activity between regulators and regulated entities
- full provenance of digital assets and full transactional history to be shared amongst participants

### **Blockchain is slow**

- Visa and Mastercard processes about 2,000 transactions per second
- Visa peak daily is about 4,000 tps and the capacity is 56,000 tps
- Bitcoin can handle about 7 transactions per second: (assuming current blocksize 1MB)
- Ethereum can do 10-20 transactions per second

Layer 2 solutions:

- Lightning Network for bitcoin <https://lightning.network>
- Raiden Network for ethereum <https://raiden.network>

### **Blockchain Programming**

Organizing a ledger:

- Bitcoin uses a ledger that just keeps track of transactions.
- Transactions specify a number of inputs and a number of outputs
- You can think of the inputs as coins being consumed (created in a previous transaction) and the outputs as coins being created

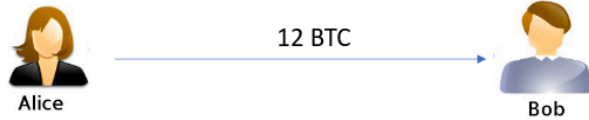
## A bitcoin Transaction:



## Bitcoin Scripting Language

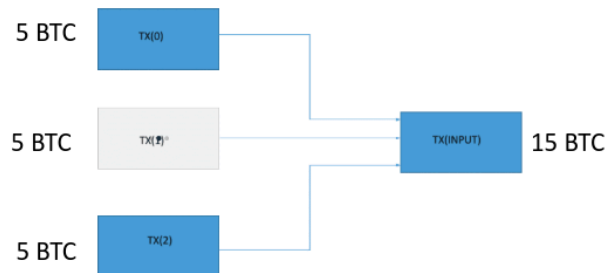
- Has a fixed set of "Op Codes" or instructions:
  - A total of 256 – 15 are disabled, 75 are reserved
  - Basic functions – arithmetic, conditionals
  - Crypto functions – hash functions, signature verifications
- Turing Incomplete
  - Does not allow infinite loops
  - Advantage: does not run malformed/malicious scripts
  - Disadvantage: does not allow for complex logic to build applications on the blockchain
- Reverse-Polish Notation
  - The operators follow operands, e.g., "1 + 2" is written as "1 2 +"
- Stack-based
  - Last-In-First-Out (LIFO)

# Bitcoin Scripts in Action



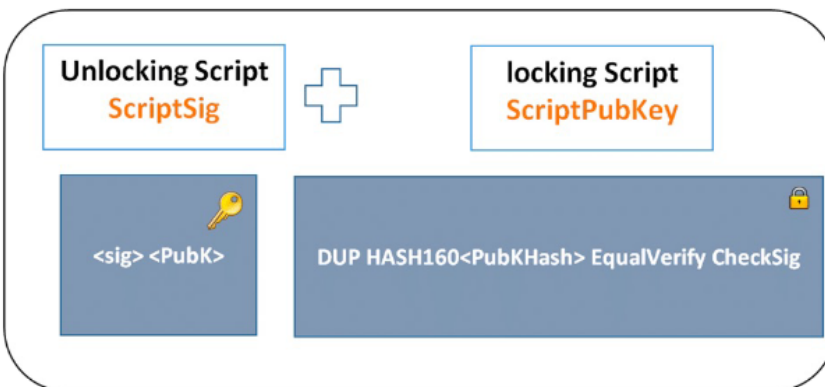
## • Transaction Input

- Alice needs to get bitcoins which she has received from various previous transactions.
- Suppose Alice needs to pull bitcoins from the following transactions which we shall name TX(0), TX(1) and TX(2)



- Alice sends Bob an output which has the scriptPubKey, which includes Bob's address.
- Bob unlocks the input using his signature of scriptSig which includes his signature and his public key.
- scriptPubKey = OP\_DUP OP\_HASH160 <Bob's public address> OP\_EQUALVERIFY OP\_CHECKSIG
- scriptSig = <Bob's signature> <Bob's public key>

## Locking and Unlocking



## Verification

- <Bob's signature> <Bob's public key> OP\_DUP OP\_HASH160 <Bob's public address> OP\_EQUALVERIFY OP\_CHECKSIG
- For OP\_DUP pop <Bob's public key> , duplicate it and push it back

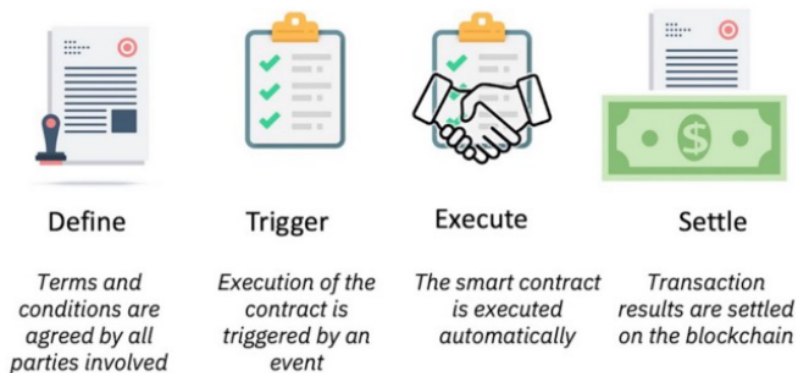
- <Bob's signature> <Bob's public key> OP\_DUP OP\_HASH160 <Bob's public address> OP\_EQUALVERIFY OP\_CHECKSIG
- OP\_CHECKSIG pops <Bob's public key> and <Bob's signature> and checks their validity.
- This is where the Elliptical Curve Digital Signature Algorithm (ECDSA) is used.

## Better Blockchain Programming Models

### Smart Contracts

- Programmatically enforced state updates
  - You can add any functionality you want!
- Can facilitate access to and distribution of funds based on specified conditions
- Can create, transfer, and alter arbitrary digital assets
- Interact with other contracts to create robust, interoperable applications
- Base layer for the Internet of Value

### How smart contracts work



## What are some advantages of a smart contracts



## **The General Theory of DApps**

- A DApp must have open-source code and work without third-party intervention. It must be user-controlled, as in, they propose and vote on changes that are automatically implemented.
- All information must be held in a publicly accessible blockchain network. Decentralization is key, as there cannot be a central point of attack.
- DApps must have some sort of cryptographic token for access, and they must reward contributors in the said token, such as miners and stakers.
- A DApp must have a consensus method that generates tokens, such as proof-of-work (PoW) or proof-of-stake (PoS).

## **Ethereum Blockchain**

- Blockchain as a Fully Distributed Database
  - Stores data
  - Transactions/messages alter the data




## AI, ML and DL

Artificial Intelligence: any technique that enables computers to mimic human behaviour

Machine Learning: Ability to learn without explicitly being programmed

Deep Learning: Extract patterns from data using neural networks

### Classes of Traditional Learning Problems

Supervised Learning	Unsupervised Learning	Reinforcement Learning
<b>Data:</b> (x,y) x is the data y is the label	<b>Data:</b> (x) x is the data No labels!	<b>Data:</b> state-action pairs and observations
<b>Goal:</b> Learn function to map $x \rightarrow y$	<b>Goal:</b> Learn underlying structure	<b>Goal:</b> Maximize future rewards over many time steps
<b>Example:</b> 	<b>Example:</b> 	<b>Example:</b> 
This thing is an apple	This thing is like the other thing	Eat this thing because it will keep you alive

Neural Networks have a kind of universality i.e., no matter what  $f(x)$  is; there is a network that can approximately approach the result and do the job! This result holds for any number of inputs and outputs.

- Implements the Universal Approximation Theorem

Given a training dataset containing  $n$  input-output pairs  $(x_i, y_i)$ ,  $i \in [1, n]$ , the goal of deep learning model training is to find a set of parameters  $w$ , such that the average of  $p(y_i)$  is maximized given  $x_i$ .

### Motivations for Democratizing AI with Blockchain

Improve models you use faster as data evolves

- Crowdsourcing: access to people + data
  - Social good: Gig economy



- Use the data collected to train models on or off-chain
- No central authority

Data is generated and stored in a decentralized way

- Cross-device: data is stored across different users' devices.
  - E.g., pictures in phones, healthcare logs in wearable devices, etc.
- Cross-silo: data is stored across different organizations (data-silos)
  - E.g., patients' data in hospitals, transaction data in banks, etc.
- These clients (data sources) constantly generate new data, with which we can train better machine learning models

**Federated Learning:** Many clients collaboratively train a model under the orchestration of a central server

- Problems:
  - Who hosts the central server
  - Malicious? Dictatorship? (price, who participates, etc.)

**Alternative – P2P Network:**

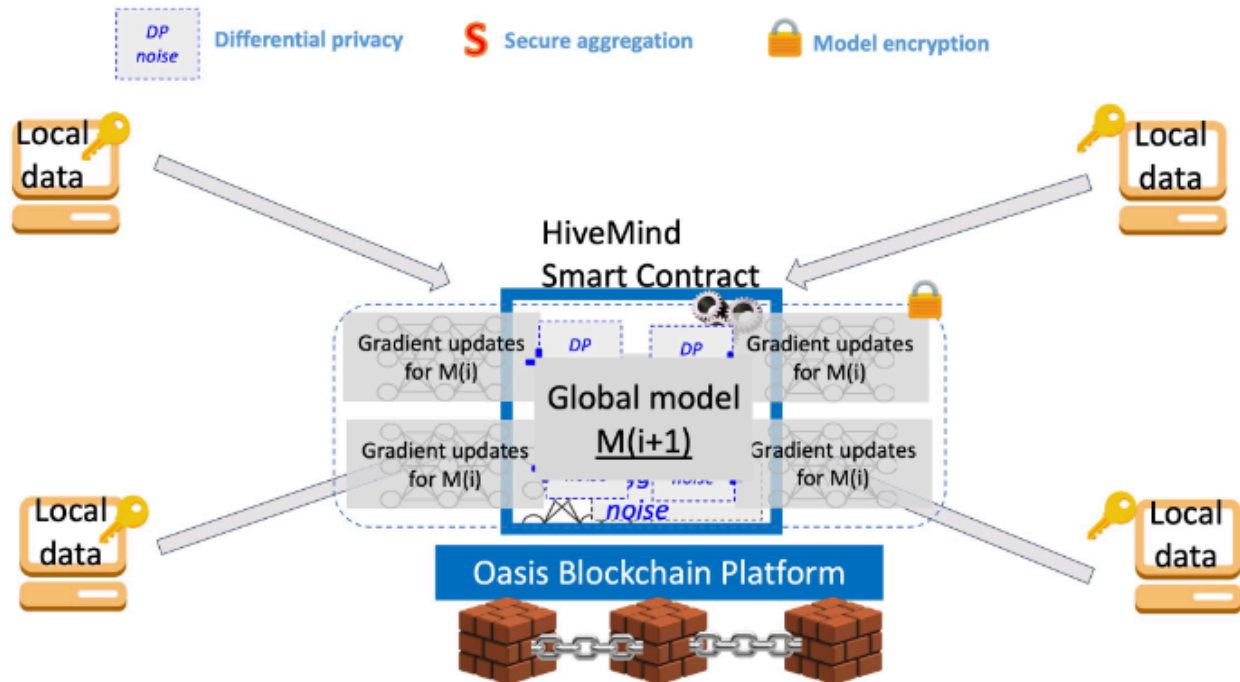
- Advantage: do not need to trust any central server
- Disadvantages:
  - Significant overhead if the number of clients is large
  - Privacy challenges

Challenge: Decentralized Federated Learning

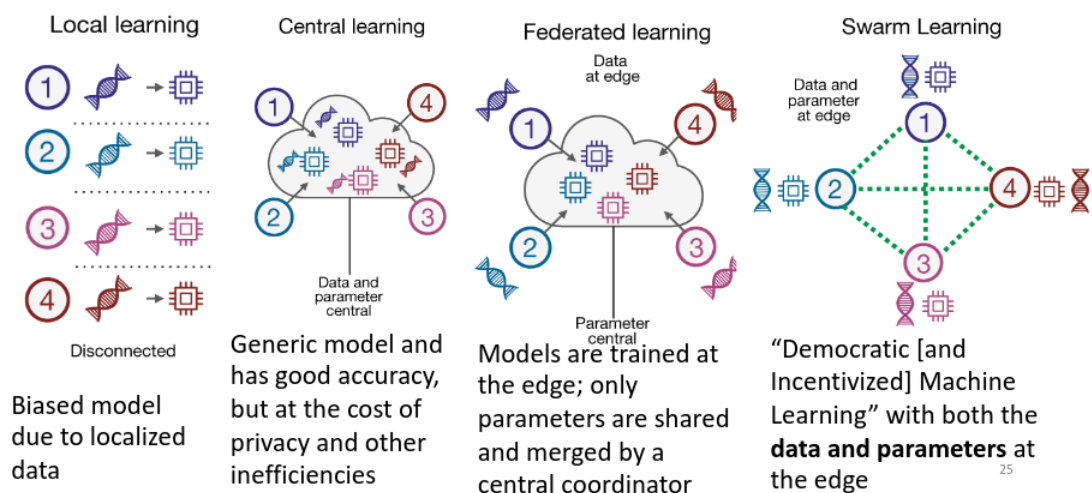
- Can we do model aggregation that satisfies the following?
  - Decentralization: Do not need to trust any central party for operations
  - Efficient and scalable: Works with a large number of participants
  - High utility: Comparable accuracy to a centralized model
  - Privacy: Protects client's data privacy
  - Good quality model inputs and outputs: Built-in incentive mechanism

- Several Proposals for Model Training Using Blockchain
  - Hivemind Decentralized Federated Learning
  - Swarm Learning
  - Sharing Updateable Models on Blockchain

## Hivemind Decentralized Federated Learning



## Different Learning Architectures

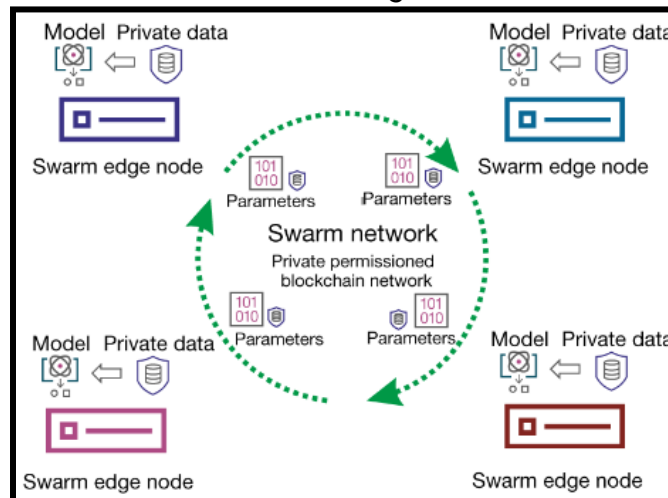


## Swarm Learning

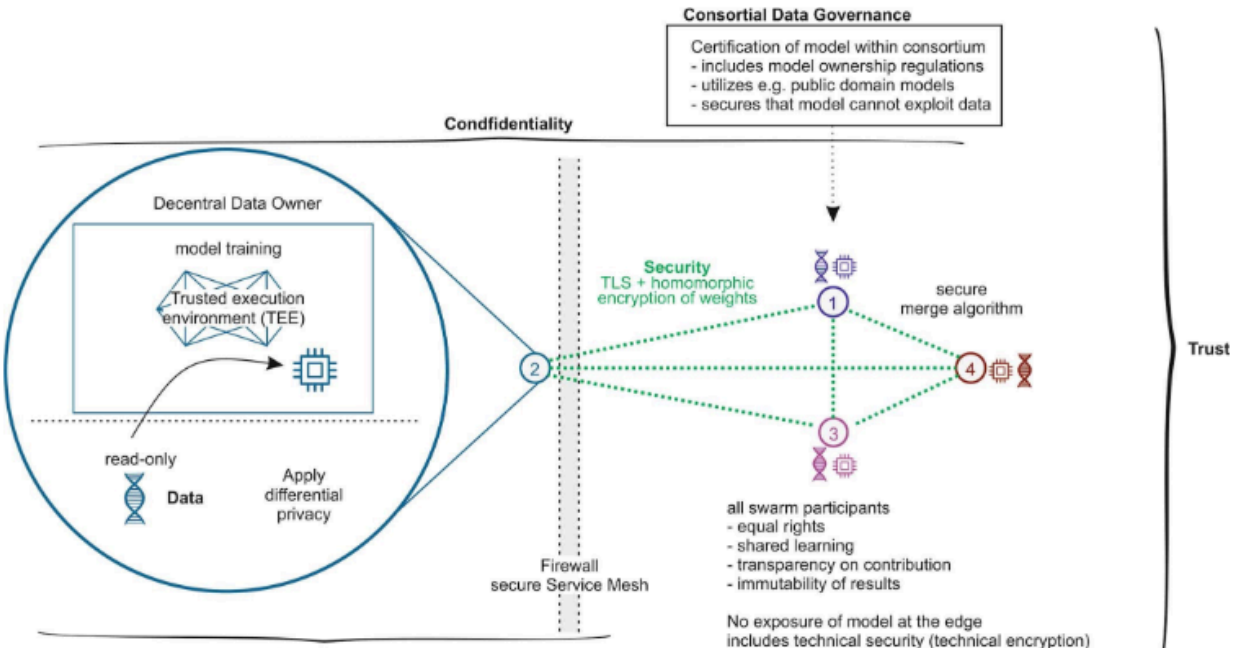
- Equal and like-minded partners in the network
- Ownership of the data remains local
- Data protection and data security are solved locally
- Less susceptible to bias in machine learning

- Workflow:

- Register
- Train
- Merge
- Repeat



## Swarm Learning Trust Model



## Sharing Updatable Models (SUM) on Blockchains

- Difficult to set up AI/ML systems
  - hardware constraints → cost constraints

- scaling
- Difficult to keep an up-to-date model available
- AI skills are centralized
  - datasets are not shared
  - charged per-query

## **Proposed System**

- 1) Deploy an initial model
- 2) Contributors submit data + deposit
- 3) Contributors can get a reward after submitting good data
- 4) The model remains free to use for inference

Ways to encourage contributors to submit good quality data.

Several examples in the paper:

1. Gamified (non-financial, points + badges like Stackoverflow)
2. Based on established theory in Prediction Markets
3. Deposit, Refund, and Take: Self-Assessment

## **GAMIFIED (NON-FINANCIAL)**

- Contribute to shared public resource (like Wikipedia)
- Points + badges like Stackoverflow
  - Milestones for number of contributions
  - Points for submitting diverse data
  - Badges for using different labels
  - Extra points for submitting data frequently
- Can be tracked on-chain or off-chain by 3rd parties

## **BASED ON PREDICTION MARKETS**

- Prediction Market: Bet or contribute a belief on the outcome of an event
  - E.g., the winner of a soccer game or an election

- Use ideas from prediction markets to incentivize good data contributions
- Phases
  1. Commitment
  2. Participation
  3. Reward

### **BASED ON SELF-ASSESSMENT**

- Predict
- Deposit
- Refund
- Take

### **Simulation**

- Assumption: “Bad Agent” frequently adds incorrect data.
- There are periodic effects because of the time required to wait before collecting a refund.
- The model can still maintain accuracy.
- Honest contributors can still profit.