Blockchain

4 key questions

What is a Distributed Ledger ?

```
In [ ]:
```

How do Blockchains work ?

In [ ]:

What is the Value of Blockchain Technologies ?

```
In [ ]:
```

What are the Main Challenges of Blockchains ?

In [ ]:

Map

1. `what`
2. `how`
3. `why`
4. Discussion
    - Opportunities
    - Challenges

1. The `what`

1.1 Distributed systems

Collections of independent computers (or nodes) working together to achieve a common goal

Coordination obtains by **messaging** over the network of nodes

The system acts as a **single coherent system**

Key dimensions:

- `fault tolerance`
- `liveness`
- `scalability`

Collections of independent computers (or nodes) working together to achieve a common goal

Coordination obtains by **messaging** over the network of nodes

The system acts as a **single coherent system**

Key dimensions:

- `fault tolerance`
- `liveness`
- `scalability`

Example: The Internet

- Goal: transfer of information (`TCP`/`IP` protocols, etc.)
- Killer application: e-mail

1.2 Distributed ledgers

A subfield of distributed systems where

Collections of independent computers (or nodes) operate a common digital record

Use cases: where maintaining a **consistent** and **immutable** record of data is **critical**.

Example: ledger of transactions

Additional key dimensions:

- `synchronization`
- `consensus`
- `tamper-proof`

A subfield of distributed systems where

> **Collections of independent computers (or nodes) operate a common digital record**

Use cases: where maintaining a **consistent** and **immutable** record of data is **critical**.

Example: ledger of transactions

Additional key dimensions:

- `synchronization`
- `consensus`
- `tamper-proof`

Example: The `Bitcoin` protocol

- Goal: transfer of token ownership
- Killer application: *to be determined...*

1.3 Elements of a DLT

Element	**Roles**
Ledger	Record storage
Users	Viewers Contributors
Nodes	Validators Recorders
Protocol	Governance Consensus

1.4 Protocol design

In distributed systems: **no single authority**.

→ The **protocol** makes **explicit rules** to ensure **consistent & immutable** record-keeping.

1.4.1 Rules for nodes & users

The protocol specifies **access & communication rights** for each participant (nodes and users).

*Who can **read/write** data*

*How to **contribute** data*

- **Identification** (`address`, `public-private key pairing`)
- **Authentification** (`signature`)
- **Authorization & Validation** (`proof-of-work`, `proof-of-stake`)

1.4.2 Rules for the ledger

The protocol specificies **consensus & recording rules** to update the ledger.

*What is **true/correct** data*

*How is data **recorded/protected***

- **Verification**: Authorization (`hash functions`) & Validity (`double spending`)
- **Consensus mechanism** (`proof-of-work`, `proof-of-stake`, `block rewards`, etc.)

1.5 Decentralisation spectrum

DLT allow for **degrees of decentralization** as a function of the rights of **nodes** and **users**:

- Visibility (`read`)
- Access (`write`)
- Liability (`identity`)

Main dimensions:

Participants	**Closed**	**Open**
Read	Private	Public
Write	Permissioned	Permissionless
Identity	Legal entity	Pseudonymous

→ `Blockhain` technologies allow for a fully **open** design of DLT:

↓

Public x Permissionless x Pseudonymous

What is the value of this specific combination?

1.6 The computer in the sky

The ideal **technical vision** for fully decentralized DLT like `blockchain` systems is a

Public good virtual machine

1.6 The computer in the sky

The ideal **technical vision** for fully decentralized DLT like `blockchain` systems is a

Public good virtual machine

- **General purpose computation**: run any software (protocol/smart contracts)
- **No physicallity nor central authority**: no shutting down nor tampering
- **Universal access**: deployment, view, contribution
- **Enforced property right**: use, exclusion, transfer

> *"This combination is the super power of the blockchain technology"*
>
> *-- Roughgarden (2024)*

1.7 First apps: crypto tokens and payments

`Bitcoin`, `Ethereum`, `Solana`, etc.

1.7 First apps: crypto tokens and payments

`Bitcoin`, `Ethereum`, `Solana`, etc.

- Public ledgers of **token ownership**
- Machines that implement protocols/smart contracts to **validate transfers**
    - No double spending, balance consistency, transfer authorization, etc.
- **No shut down button**
    - No physicallity nor central authority
- **Universal access** on all sides
- **Property rights**
    - Self-custody of keys and related assets

2 The `how`

Mix of **cryptographic** technology and **economic** design.

- **Crypto**: how to guarantee records are
    - public
    - pseudonymous
    - tamper-proof
- **Economic**: how to incentivise nodes to
    - produce valid records
    - promote valid records

2.1 Crypto guarantees

Transaction records require:

- An origin and a destination
- Authorization

The ledger guarantees

- Anyone can freely create an identity (=address)
- Everything is public and verifiable
- Historical records are immutable

Transaction record

	Origin	Destination	Amount	Authorization	(other)
true information	Tarik	Nikolas	5 BTC	Signature (Tarik)	fee, message, etc.
public information	x012jdoijdpoj..	x32ffwew..	5 BTC	Signature (x012jdoijdpoj..)	etc.

Properties:

- Pseudonymity
- Signature
- Immutatbiliy

We obtain this using **elements of cryptography**

1. `Key pairs`
2. `Public signatures`
3. `Hash functions`

This is the secret sauce

2.1.1 Key pairs

In **public-key cryptography**, there are 2 keys:

1. `PrivKey` : the **private key**
2. `PubKey` : the **public key**

such that

```
PubKey = G . PrivKey
```

Properties:

- **Costless verification**: `PubKey` can be easily derived from the `PrivKey` .
- **Pre-Image resistance**: Given `PubKey` , impossible to obtain `PrivKey`

What is an identity in a blockchain?

- A user's identity is an **address** which obtains from the **public key**.
- Key pairs can be **freely** generated through **asymmetric encryption**

```
In [7]: generate_key_pair()
```

```
Private Key: 2e2618bbd9087d6d8b460c3bfce731610fcd031b02a99b0067
1f041f0c706774
Public Key: cb13730df21015340f89b11fa53dc29d3ec5889e04696750da7
81bec434896636097d7ed5f2954aafa7ab2e936dd239374ddc3ab84349bd0f6
bd1a911569bb67
```

2.1.2 Public signatures

To guarantee **communication** in an <u>open ecosystem</u> (**universal access and pseudonymity**), `blockchain` protocols imply that:

- **Messages** (transactions) need to reference an adresses via **signatures**
- Signatures need to be **verifiable**: everyone can **confirm the validity** of a signature

To guarantee **communication** in an open ecosystem (**universal access and pseudonymity**), `blockchain` protocols imply that:

- **Messages** (transactions) need to reference an adresses via **signatures**
- Signatures need to be **verifiable**: everyone can **confirm the validity** of a signature

Crypto signatures rely on 2 functions:

1. `Sign()`
2. `Verify()`

Given a message `x` (transaction):

- The **private key** signs the message

```
Signature = Sign(x, PrivKey)
```

- The **public key** is used by others to verify that the signature came from the owner of the private key.

```
IsValid = Verify(x, Signature, PubKey)
```

2.1.3 Hash functions

A **hash function** is a mathematical object $H()$ that takes an **input** (x) and returns a **deterministic fixed-size string of bytes** which appears **random** (h - `hash`).

```
In [8]: hash_function_example_1()
```

```
first input     monday november 17 2025

Input 1: monday november 17 2025
Hash 1 : bf6d4edb9eede8508863efea46e2765947b16fc39f449d18301c51
1b872d0fb4
```

Hash properties

$$H(x) = h$$

Property	Description
Fixed size & pseudo-randomness	Hash outputs have a fixed length and appear random, regardless of input size.
Costless verification	Given x, it is easy to compute $H(x)$.
Pre-image resistance	Given h, it is infeasible to find any x such that $H(x) = h$.
Collision resistance	It is infeasible to find two distinct inputs x and y such that $H(x) = H(y)$.
Avalanche effect	A small change in input (e.g., flipping one bit) results in a drastically different output $H(x)$ vs $H(x\|e)$

In [10]:

```
hash_function_example_2()
```

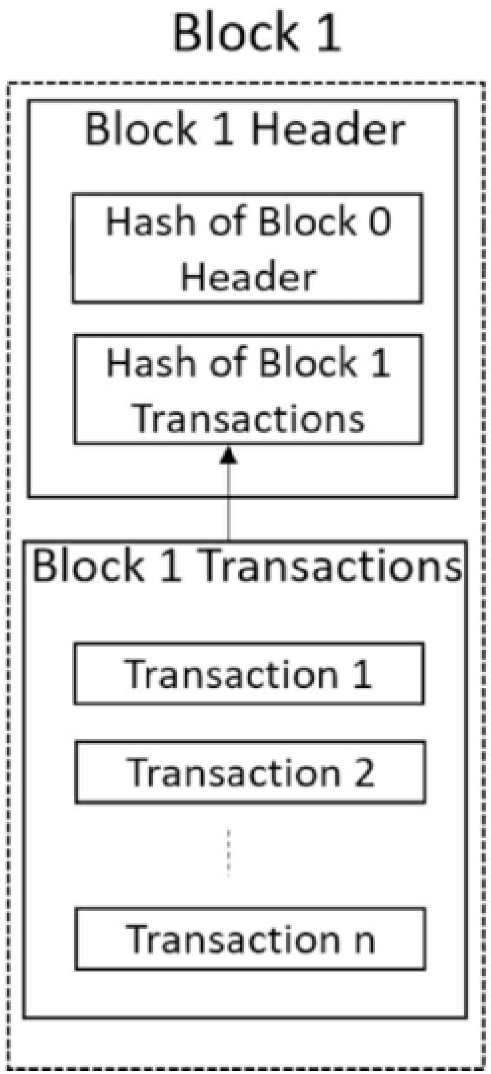
```
first input     monday november 17 2025monday november 17 2025m
onday november 17 2025monday november 17 2025monday november 17
2025monday november 17 2025monday november 17 2025
second input    monday november 17 2025monday november 17 2025m
onday november 17 2025

Input 1: monday november 17 2025monday november 17 2025monday n
ovember 17 2025monday november 17 2025monday november 17 2025mo
nday november 17 2025monday november 17 2025
Hash 1 : 2655685559e3eda77a122f03a2ac73b9061fc40bc327ec27ff8b3c
```

Hashing to guarantee immutability

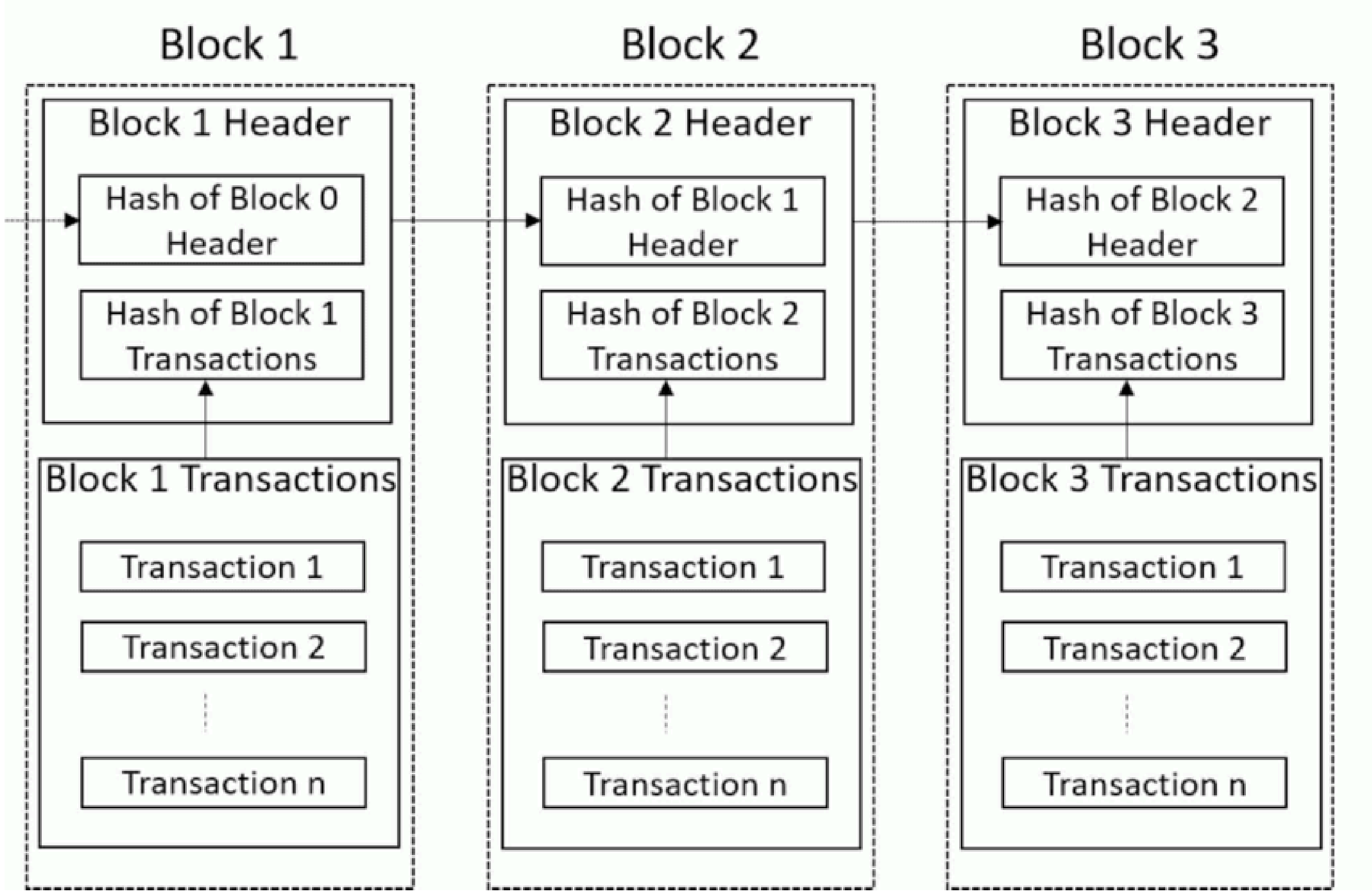
A **blockchain** is made up of a chain of **blocks**, where each block contains a list of records and a reference (or link) to the previous block.

→ The references and the data are always accompanied with their **hash**.

Block structure



Chain of blocks



Properties of Blockchain

Property	Description
Tamper-Proof by virtue of **hashing**	Each block’s hash depends on its content and the previous block’s hash. The **avalanche effect**: infinitesimal changes in one block alter the entire chain.
Public access by virtue of **crypto keys**	Activity is recorded as entries signed by **public keys**. Anyone can verify signatures, ensuring transparency and auditability.
Privacy & private ownership by virtue of **crypto signatures**	Only the holder of the **private key (PrivKey)** can initiate new activity from a **public address**, ensuring control and ownership.

2.2 Economic guarantees

How to ensure new records are valid?

- Special participants
- Vulnerabilities
- Consensus
- Incentives

2.2.1 Special participants: Validators / Miners

In a blockchain network, **validators** are special participants **responsible for maintaining the integrity and security of the blockchain**. Their role is essential for the validation and proposal of new blocks.

2.2.1 Special participants: Validators / Miners

In a blockchain network, **validators** are special participants **responsible for maintaining the integrity and security of the blockchain**. Their role is essential for the validation and proposal of new blocks.

Workflow

1. **Collect valid records** and propose **blocks**
2. **Validate other blocks**
3. **When consensus: update the ledger**
    - The proposed block is added to the blockchain
    - Validators add it to their local copy of the blockchain.

Example of record validity in payments

A transaction is **valid** if:

1. Transfer between **existing** addresses
2. **Amount** under the balance of the originator (**no double spending**)
3. **Authorized** (signature) by the originator

2.2.2 Blockchain vulnerabilities

Pseudonymity and **universal** access at **zero-cost** to become a **validator** introduces **critical vulnerabilities**:

- **Malicious or faulty behavior**: `Byzantine fault tolerance`
- **Majority control**: `Sybil attacks`

These are the main barriers to achieving correct consensus.

Pseudonymity & `Byzantine Fault Tolerance`

The **Byzantine General Problem**: If some validators are **faulty** or **malicious**, how to achieve consensus and guarantee truth?

`Byzantine Fault Tolerance` **(BFT)** refers to a blockchain's ability to maintain network functionality and consensus even when some participants behave improperly.

Zero-cost access & `Sybil Attacks`

In a **`Sybil attack`**, an adversary influences the network through **fake identities** (**`Sybil` nodes**) to overwhelm honest participants.

2.2.3 Consensus

2.2.3 Consensus

`Blockchain` systems like `Bitcoin` and `Ethereum` are exposed to issues of **`Byzanting Fault Tolerance`** and **`Sybil Attacks`**.

They therefore implement specific **consensus mechanisms** designed to handle such potential failures.

→ Goal of a consensus protocol: ensure that all participants have the same version of the ledger while addressing the vulnerabilities of permissionlessness.

General fix: leverage **costly actions** to make **attacks** on the ledger **too expensive** to undertake:

- **Selection of a validator** to propose a block conditional upon some **cost** faced by the validator (ex-ante / ex-post)
- **Rewards to a validator** from proposing a block conditional upon **references from blocks downstream**

Intuition:

- Prevents `Sybil attacks` by making participation in validation costly, rendering it infeasible for an attacker to create enough fake identities to influence the network (`51% attack`).

- Ensures `Byzantine Fault Tolerance` by making it economically prohibitive to attach new blocks to faulty records.

Examples

- **Proof of Work (PoW)** (`Bitcoin`)
    - **Costly action**: validators (**miners**) solve **computationally difficult cryptographic puzzles** before they can propose new blocks.
    - The **difficulty** is adjusted to the size of the miner market.
    - **Energy Consumption**: miners expend significant computational resources (and electricity) to solve the cryptographic puzzles.
    - **Risk of Centralization**: small number of miners with large computational power could have a disproportionate influence over the network.

- **Proof of Stake (PoS)** (`Ethereum 2.0`)
    - **Validators** "stake" their wealth as collateral to back their choices.
    - **Selection** to propose and validate new blocks based on a **validator's staked amount.**

      In case of **misconduct**, stakes are **slashed**.
    - **Energy Efficiency**: unlike PoW, PoS does not require validators to perform computational work, thereby reducing energy consumption.
    - **Risk of Centralization**: small number of validators with large stakes could have a disproportionate influence over the network.

2.2.4 Economic incentives

Validators/miners **invest** time, computational resources, and capital into maintaining the blockchain.

Sources of revenue

- **Block rewards**: minting of new tokens allocated to successful block creators.
  → Algorithmic (deterministic) monetary policy
- **Transaction fees**: fees paid by users when submitting a transaction on the blockchain. Awarded to the validator who successfully adds the block to the blockchain.
  → Manage network congestion
  → Prevent spamming and DoS attacks

3 The `why`

Issues with centralized alternatives

While centralized systems have been the backbone of many industries - in particular information based economies -, they come with significant challenges that distributed systems like blockchain seek to overcome.

Primary issues of centralization:

- **Data Manipulation**:
- **Market power**
- **Entry barrier**
- **Lack of innovation**

Ulimately, the key is the **governance of information** which derives from **transparence** and **trust**.

More on this in the DeFi Lecture

The blockchain trilemma

The blockchain trilemma

The **Blockchain Trilemma** is the **inherent trade-offs** between three key properties of blockchain networks:

1. **Decentralization**: distribution of control and decision-making
2. **Security**: guarantee for integrity and reliability of records
3. **Scalability**: capacity to scale without degrading performance

The trilemma states that blockchain networks can optimize for **only two of these three properties** simultaneously without compromising the third.

Examples

- **Security vs. Scalability**: To enhance scalability, a blockchain might increase its block size or reduce block confirmation times. However, this can lead to weaker security, as nodes have less time to validate transactions, increasing the risk of errors or malicious behavior.

- **Scalability vs. Decentralization**: To improve scalability, some networks may rely on fewer, more powerful nodes to increase transaction throughput. This, however, reduces decentralization by concentrating control in the hands of fewer participants.

- **Decentralization vs. Security**: Increasing decentralization can lead to network delays and inefficiencies, making it harder to implement strong security measures that require rapid consensus among many nodes.

In search for the killer app

Blockchain is a **general-purpose technology**, and the full potential of its applications is still being explored.

Candidates for blockchain’s "killer apps":

- **Private money**
- **Initial Coin Offerings (ICO)**
- **Stablecoins**
- **Automated Market Makers (AMMs)**
- **Non-Fungible Tokens (NFTs)**
- **Decentralized Autonomous Organizations (DAOs)**

Beyond finance: the promise of Web3

- **Web3** is a decentralized version of the internet, built on blockchain technology, that promises to return control of data, privacy, and value to the users rather than centralized corporations.
- The promise of **Web3** lies in its ability to create an internet that is more **user-owned, decentralized, and transparent**, fostering a new wave of innovation and empowering individuals with more control over their digital lives.

Discussion

The evolving technological landscape

The `How`

Development of infrastructure:

- **Blockchain Trilemma**: Balancing **decentralization**, **security**, and **scalability**. Innovations like Layer 2 solutions, sharding, and consensus protocol improvements.
- **Hash properties**: continuous update of hash function to ensure properties like pre-image resistance are not violated.
- **MEV (Miner/Maximal Extractable Value)**: Understanding how miners (or validators in PoS systems) can extract additional value through transaction ordering. This can have implications for market fairness and transparency.
- **Interoperability Solutions**: Update on technologies that allow different blockchains to communicate, such as Polkadot, Cosmos, and bridges.
- **Privacy-Preserving Technologies**: Zero-knowledge proofs (e.g., zk-SNARKs) and confidential transactions. Impact on anti-money laundering (AML) and compliance.

The `Why`

Development of killer applications:

- **Payments**: Blockchain's potential to streamline cross-border payments, reduce settlement times, and lower costs.
- **Stablecoins**: Their role in providing digital representations of fiat currencies. Implications for monetary policy and financial stability.
- **Exchanges**: The rise of decentralized exchanges (DEXs) like Uniswap, which operate without intermediaries and provide automated liquidity.
- **Non-Fungible Tokens (NFTs)**: How digital ownership and asset tokenization can affect market behaviors, intellectual property, and digital commerce.