Blockchain Basics

Bitcoin is Turing incomplete.

Ethereum is Turing complete.

Oracle Problem

Blockchain cannot interact with external world data directly due to security and immutability but in some cases, we need real world data in real time like stock prices or payment information.

Oracles as the Bridge: Oracles are entities that bridge this gap by providing external data to the blockchain.

They act as intermediaries between the blockchain and the real world.

Chain-link is a decentralized oracle network that bridges the gap between blockchain technology and the real world. Essentially, it provides a secure and reliable way for smart contracts to access external data and fulfill off-chain computations.

What is Layer 2?

Layer 2 solutions are essentially secondary frameworks or protocols built on top of an existing blockchain (Layer 1) to enhance its scalability, speed, and cost-efficiency.

There are two types of true layers 2:

1. Optimistic Rollups
2. Zero Knowledge Rollups

Dapp → Decentralized Application.

Web1 → The permissionless open-sourced web with static content.

Web2 → The permissioned web with static content.

Web3 → The permissionless web with dynamic content.

Genesis Block: The first block in blockchain is called Genesis Block. The previous hash of this block does not exist or is 0.

Mining: The process of finding solution to the “Blockchain” problem. In start the “problem” was to find a hash that starts with four 0’s. Now, the number of 0’s in the start have increased to increase the complexity of the blockchain.

Block: A list of transactions mined together.

Decentralized: Having single point of authority.

Nonce: A number used once to find the “solution” of the blockchain problem.

Private Key: Only known to key holder, it’s used to “sign” transactions.

Ethereum runs on keccak256 hashing algorithm.

Consensus: Consensus is the mechanism used to agree on the state of the blockchain.

Consensus can be broken down into two pieces.

1. Chain Selection Algorithm.
2. Sybil Resistance.

Sybil Resistance: is a blockchain's ability to defend against users creating a large number of pseudo-anonymous identities to gain a disproportionately advantageous influence over said system.

In simple terms, it is protection against someone against somebody creating a bunch of fake blockchains so that they can get more and more rewards.

There are 2 types of Sybil Resistance mechanisms:

1. Proof of Work.
2. Proof of Stake.

Proof of Work: In this proof, you need to go through a very tough riddle which needs very high computational power to solve to provide a nonce which can lead to the hash having four 0's in the starting.

> **Note:** Some blockchains may make their riddle or their block answer intentionally hard, or intentionally easy to adjust the block time - which is the average time it takes to mine a block. Block time is proportional to how difficult these algorithms are.

Proof of Work needs to be combined with a chain selection rule to create consensus.

A **chain selection rule** is implemented to determine which blockchain is the *real* blockchain. Bitcoin (and prior to the merge, Ethereum), both use something called Nakamoto Consensus. This is a combination of Proof of Work (Ethereum has since switched to Proof of Stake) and the longest chain rule.

In the longest chain rule, the decentralized network decides that whichever chain has the greatest number of blocks will be the valid, or *real* blockchain

You'll sometimes hear people use **Proof of Work** to describe a consensus mechanism, but it's a little bit inaccurate, it's really the combination of sybil resistance *and* chain selection that create consensus

Reward: The first block to figure out riddle to the problem is rewarded in two ways.

1. Transaction Fees
2. Block Reward.

Bitcoin reward is halved after every 4 years.

Sybil Attack: When a user creates several pseudo-anonymous accounts to try to influence a network.

51% Attack: Occurs when a single entity possesses both the longest chain and majority network control. This would allow the entity to fork the chain and bring the network onto the entities record of events, effectively allowing them to validate anything.

Cons of POW: POW uses a lot of computational power which highly impacts the environment.

Proof of Stake

In contrast to trying to solve a block problem, Proof of Stake nodes put up some collateral that they are going to behave honestly aka they stake. If a node is found to be misbehaving, it's stake is slashed. This serves as a very effective sybil resistance mechanism because for each account, the validator needs to put up more stake and misbehaving risks losing all that collateral.

In a Proof of Stake system, miners are known as **validators**. They aren't mining blocks they're just validating other nodes.

Unlike in Proof of Work, where each node is racing to solve the block problem first, in Proof of Stake, validators are pseudo-randomly chosen to propose the next block and other nodes will validate it.

Proof of Stake of course comes with its own Pros and Cons.

Pros:

- great sybil resistance mechanism
- great for the environment, much less energy

Cons:

- seen as less decentralized due to upfront staking costs

Layer 1 and Layer 2

Layer 1 solutions: This refers to base layer blockchain implementations like Bitcoin or Ethereum.

Layer 2 solutions: These are applications added on top of a layer one, like Chainlink or Arbitrum.

➔ Sharding and Rollups are scalability solutions.

L2 or Layer-2 are most of the times called as **Rollups.**

What is a Roll up?

Scaling Solution that increases the number of transactions on L1 chain without increasing the gas fees.

Why we need Rollups?

Rollups help solve the blockchain trilemma, which states that a blockchain can only achieve two out of three properties: *decentralization*, *security*, and *scalability*. In the case of Ethereum, **scalability** is sacrificed as it can only process approximately 15 transactions per second. Rollups, on the other hand, aim to enhance scalability without compromising security or decentralization.

How Rollups Work

When a user submits a transaction to a rollup, an **operator** (a node or entity responsible for processing transactions) picks it up, bundles it with other transactions, compresses them, and submits the batch back to the L1 blockchain. This process allows for efficient handling of transactions as gas costs associated with the transaction, are split among all the users that submitted the transactions in the batch.

There are two types of rollups, Optimistic and Zero-Knowledge rollups. The main difference between the two lies in how each rollup verifies the validity of the transactions.

Optimistic Rollups

They assume that off-chain transactions are *valid by default*. Operators propose the **valid state** of the rollup chain, and during a **challenge period**, other operators can challenge potentially fraudulent transactions by computing a **fraud proof**.

This **fraud proof process** involves the operator engaging in a *call and response interaction* with another operator to identify and isolate a specific computational step. This specific step is then executed on the Layer 1 blockchain: if the result differs from the original state, it indicates that the transaction was fraudulent. When the fraud proof succeeds, the rollup will re-execute the entire batch of transactions correctly, and the operator responsible for including the incorrect transaction will be penalized, usually by **losing staked tokens (*slashing*).**

Zero-Knowledge (ZK) Rollups

ZK rollups use validity proofs, known as *zk proofs*, to verify transaction batches. In this process, the **prover (operator)** generates a zk proof to show that their inputs (the transactions) satisfy this equation. A **verifier (an L1 contract)** then checks this proof to ensure that the output matches the expected result. The solution that the prover uses to demonstrate that their input satisfies the mathematical equation in the zk proof is commonly referred as the **witness.**

What is Sequencer?

The role of a **sequencer** is crucial for ordering and bundling transactions. Sequencers are operators that are responsible for organizing how transactions are processed. In many roll-up solutions, sequencers are centralized, controlled by a single entity.

Cons of Sequencer Being Centralized:

1. Manipulation & Censorship
2. Operational Downtime

A Layer 2 (L2) chains maturity is evaluated based on specific properties and categorized into **stages**.

The L2B team provides an opinionated assessment to encourage a progression towards a greater decentralization.

Rollup Stages

1. **Stage 0:** In this initial stage, the rollup's governance is largely in the hands of the operators and a security council, ensuring that critical decisions and actions are overseen by a *trusted group*. The open-source software allows for the reconstruction of the state from L1 data, ensuring transparency and accessibility. Users in this stage have an exit mechanism that allows them to leave the rollup within seven days. However, this often requires actions from an entity/operator.
2. **Stage 1:** In this stage, governance evolves to be managed by *smart contracts*, although the *security council* still plays an important role (e.g. solving bugs). At this stage, the proof system becomes fully functional, enabling decentralized submission of validity proofs. The exit mechanism is improved, allowing users to exit independently without needing operator coordination.
3. **Stage 2:** In this final stage, the rollup achieves full decentralization with governance entirely managed by smart contracts, removing the need for operators or council interventions in everyday operations. The proof system at this stage is permissionless and the exit mechanism is also fully decentralized. The security council's role is now strictly limited to addressing any errors that occur on-chain, ensuring that the system remains fair without being overly reliant on centralized entities.

Bridging: Taking the funds from one chain and getting them to another chain.

Finality: This term refers to the time from sending the transaction to when it is considered settled. On Ethereum, this takes about 13 minutes, but on zkSync it can take approximately 24 hours. During this period, transactions are displayed **instantly** in the UI and can be further transferred, but full finality should be awaited to ensure they are fully received and validated using ZK proofs.