From Zero to zkEVM - Road to Scalability

v.1.0

December 11, 2023

1 Document Pre-requisites

A comprehensive understanding of the following key areas is necessary to fully follow this document:

- **Basics of the Ethereum L1 Execution Layer:**
  - How Smart Contracts work.
  - Basics about Token Smart Contracts (mainly ERC20).
- **Basic cryptographic concepts:**
  - Digital signatures.
  - Hashes and Merkle Trees.
- **Programming Languages:**
  - The zkEVM node is written in **Golang**.
  - The executor and the prover are written in **C++**.
  - The smart contracts are written in **Solidity**.
  - While not the primary language for development, a basic understanding of **JavaScript** is necessary to follow this document.

2 Blockchain Scalability

2.1 Introduction

A **blockchain** is a decentralized and peer-to-peer network that employs cryptographic techniques to securely host applications, store data, and facilitate the seamless exchange of digital assets that represent real-world currency.

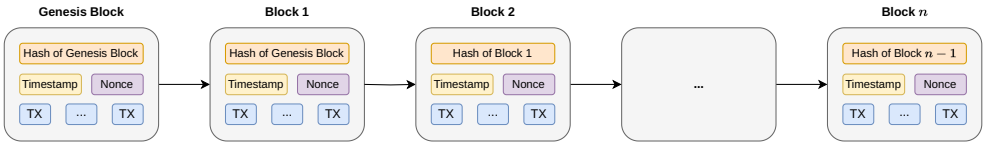

Figure 1: Visual representation of a Blockchain.

The issue of **blockchain scalability** emerges as a foundational challenge within the realm of decentralized blockchain networks, and it undoubtedly stands as the most significant bottleneck in their development. This problem is related with the capacity and efficiency of these networks as they grow in terms of user participation and transaction volume. More specifically, as more users and transactions are added to a blockchain network, its ability to process and validate these transactions in a timely and cost-effective manner becomes increasingly restricted.

The term *scalability trilemma* was first coined by Vitalik Buterin to describe the inherent tension between three properties that a high-performing blockchain platform must have: **decentralization**, **security** and **scalability**.

1. **Decentralization:** This denotes the blockchain's capacity to operate without relying on trust relationships with a limited group of highly centralized entities.

2. **Security:** This represents the blockchain's capability to withstand attacks from a substantial proportion of participating nodes, ideally around 50%.

3. **Scalability:** This refers to the blockchain's ability to augment the amount of processed transactions per second (TPS), increasing the overall system throughput.

The *trilemma* refers to the belief that blockchain platforms can only achieve two of these three goals effectively.

Blockchain scalability is a critical concern because it impacts the network's ability to maintain decentralization, security, and usability while accommodating a growing user base and transaction load. Addressing this problem is essential for realizing the full potential of blockchain technology.

2.2 Blockchain Scalability Strategies

Intuitively speaking, we can conceptualize an **EVM (Ethereum Virual Machine) blockchain layer 1** as a set of states, each of them encapsulating the account balances, nonces, bytecode for smart contracts, storage values, and more. After **executing** a block of transactions at state S_i, a new state S_{i+1} appears. We will look into **Ethereum World State** in more detail later on, but it is the root of a Merkle Tree containing data for each of the Ethereum accounts. This state creation iterative process serves as the foundation for constructing a blockchain.

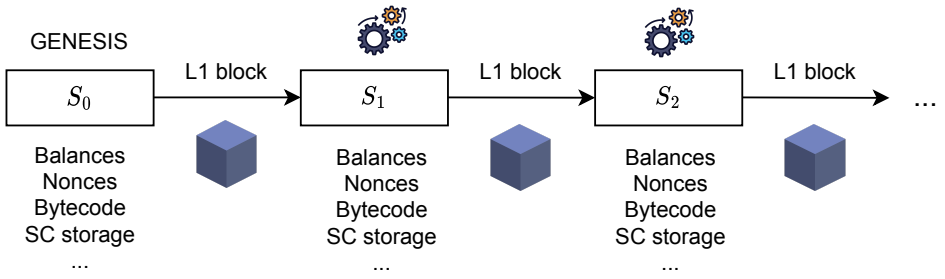

Figure 2: Blockchain as Sequence of States.

In this scenario, transactions are both **available** to the entire network, ensuring visibility for all participants, and are **executed** by the EVM runtime environment. These transactions are incorporated into blocks, and smart contracts define the rules for the **execution** of these blocks.

More transactions per block The initial strategy we can employ to enhance blockchain scalability involves increasing the number of transactions per block. This approach carries inherent risks that demand careful consideration. More specifically, as the volume of transactions per block surges, a significant number of blockchain nodes may find themselves stretched to their resource limits. Consequently, this escalation in resource demand can potentially lead to centralization, where the network becomes dominated by powerful nodes.

Sharding Another strategy for scaling consists in the division of the network's workload into discrete segments, known as shards. In this approach, each node within the blockchain network is assigned responsibility for managing a specific shard. Consequently, a node only handles the operations and transactions relevant to that particular shard. Within the sharding strategy, there exist several approaches:

(a) The initial approach involves using the existing L1 blockchain as a consolidation chain, which will be subsequently designated as the L1 **beacon chain**. In this approach, the beacon chain servers as a central coordinating entity.

Individual shards, which themselves function as distinct chains within the network, are tasked with executing a subset of blocks. On one hand, each shard ensures availability, guaranteeing that transactions are accessible This accessibility extends to all nodes operating within the shard, all of which require access to transaction data.

On the other hand, each shard possesses the execution capability akin to that of the original Layer 1 chain (the ability to process transactions, deploy smart contracts, and perform other essential blockchain operations).

In essence, this approach creates a scenario in which both availability and execution are inherent features of each individual shard.

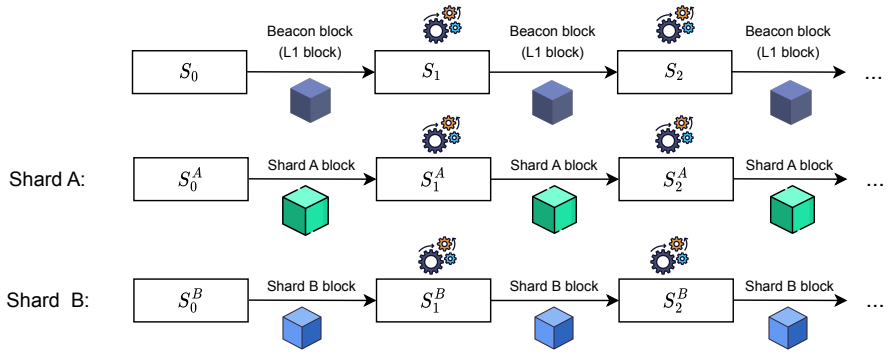

Figure 3: Schema of the initial sharding approach.

Nonetheless, this approach gives rise to significant challenges within the Ethereum Layer 1 specifications. Firstly, L1 Ethereum assumes the responsibility for managing the states of each individual shard. This entails the execution of transactions and facilitating inter-shard communication when required. Henceforth, an inherent consequence of this approach is the potential hindrance to L1 ossification. In the context of Ethereum, ossification refers to the stabilization and immutability of the Layer 1 protocol. The need for L1 Ethereum to actively manage and adapt to the evolving requirements of each shard can impede the process of ossification, as it requires ongoing updates and adjustments to accommodate the dynamic nature of the network. In conclusion, this method generates a huge instability in L1.

(b) In this approach Ethereum specifications provide a single L1 execution layer providing execution, maintaining a unified L1 execution layer responsible for transaction processing and ensuring the accessibility of data, together with a data availability sharding scheme (EIP-4844: Shard Blob Transactions). So, the other shards are data shards, they can be defined as chains of data availability without the capacity of executing transactions.

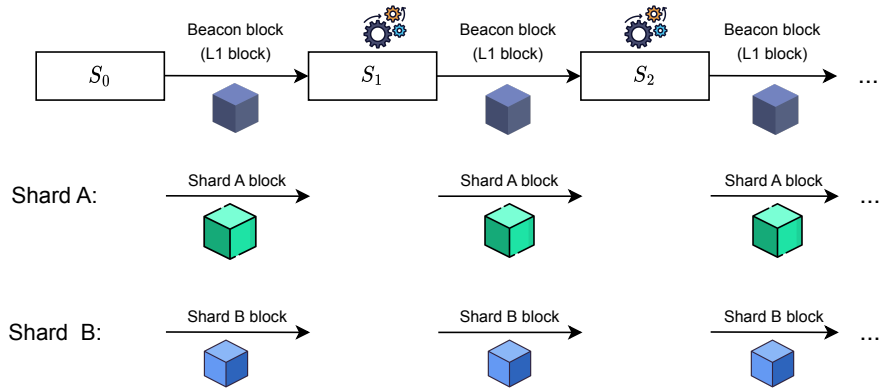

Figure 4: Schema of the post EIP-4844 design.

In this scheme, instead of conventional transactions, the blocks within each shard are now structured to contain **blobs** (binary large objects). These binary data hold no intrinsic meaning for the Layer 1; their intended purpose is to be decoded and processed by upper layers, which are widely known as **Layers 2**.