Optimistic and Zero-Knowledge Rollups

Understanding blockchain

Angelo PICERNO

Eduardo TEIXEIRA DE SOUSA

Elena PEROTTI

Micaela MASRI

Adrien SEIGLE

November 2025

Contents

1. Introduction

The emergence and widespread adoption of blockchain technology has revealed a fundamental structural limitation: scalability. Along with decentralization and security, scalability is one of the three fundamental components of the blockchain trilemma. The trilemma states that a blockchain system can only optimize two of these three properties at the same time, forcing designers to accept trade-offs in performance or trust assumptions.

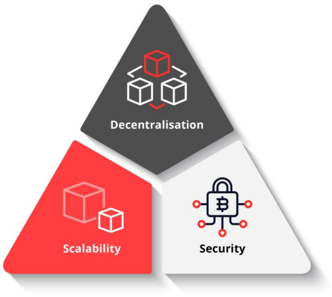

Figure 1: The blockchain trilemma

Interestingly, scalability challenges are not unique to blockchain systems. Traditional database architectures have long faced a comparable tension, often described as the CAP Theorem. Even though applied to different technologies, both frameworks highlight the same underlying idea: achieving perfect performance, resilience and correctness at the same time is structurally difficult, and often impossible.

Blockchain scalability can thus be understood not as an isolated limitation, but as the continuation of a broader historical challenge in distributed systems.

There are two main strategies to improve blockchain scalability. The first one is to modify the underlying blockchain architecture (Layer1, or L1). The second is to implement a second layer (L2) consisting of auxiliary technologies built on top of L1.

Among L2 solutions, Rollups (and in particular Optimistic Rollups and Zero-Knowledge Rollups) have gained central importance because they shift most computational load off-chain, while preserving the security guaranteed by Layer 1.

This document examines the technical and functional characteristics of both approaches and discusses their differences, advantages and limitations, as well as potential adoption scenarios. The goal is to assess how these mechanisms contribute to addressing the scalability challenge and to evaluate the extent to which they can offer a viable path toward more efficient and widely deployable blockchain systems.

2. Technical Background

In this section, we introduce the fundamental concepts necessary to understand blockchain scalability. We will review the key components of blockchain architecture, the challenges inherent to distributed ledger systems, and the mechanisms that have been proposed to enhance transaction throughput without compromising security. This background will provide the foundation for the subsequent discussion of Layer 2 solutions and Rollups.

2.1. Scalability and Layer 1

A Layer 1 blockchain is a foundational blockchain that processes and validates transactions independently within its own network, without relying on external systems. Examples of Layer 1 blockchains include Bitcoin, Ethereum, Solana, Aptos, and NEAR.

Several solutions have been proposed to increase the transaction processing capacity at the Layer 1 level. These include *sharding*, which enables parallel validation of transactions across different shards, and modifications to the consensus protocols to make them more efficient.

2.1.1 Sharding

Sharding consists of dividing a blockchain into smaller, more manageable segments, called shards, which can process transactions and execute smart contracts in parallel. By distributing the computational load across multiple shards, blockchain networks can significantly improve throughput while reducing latency and resource consumption.

Unlike traditional blockchain architectures where all nodes must process every single transaction, sharding allows subsets of nodes to validate only a portion of the total transactions, thus increasing scalability.

For example, if a network of 1,000 nodes is divided into 10 shards of 100 nodes each, the transaction processing speed can increase by approximately a factor of ten.

An example of a sharding implementation is the NEAR Protocol, a Layer 1 blockchain that uses Nightshade. Its network is capable of processing up to 160,000 transactions per second (TPS), outperforming many other blockchain platforms.

While sharding improves scalability, it also introduces potential security risks. If a shard contains too few nodes or if a single shard is targeted by malicious actors, it may become easier to compromise the consensus within that shard. Therefore, blockchain designers must carefully balance shard size, the total number of shards, and mechanisms for cross-shard communication to maintain the overall security of the network.

2.1.2 Consensus Protocols: PoW vs. PoS

Another approach to improving blockchain performance is the adoption of alternative consensus protocols, which affect transaction speed, security, and energy efficiency.

Proof of Work (PoW) is a transaction validation algorithm based on solving computational problems, requiring network participants (miners) to perform complex cryptographic operations. PoW-based networks have significant environmental costs due to the high energy consumption associated with mining. Additionally, the need for specialized, high-performance hardware further increases the economic burden of maintaining network security.

An alternative consensus protocol, **Proof of Stake (PoS)**, does not require high computational power and significantly reduces transaction times, making it a more efficient solution compared to PoW. In PoS, instead of solving cryptographic puzzles to find the correct nonce, users must prove ownership of a certain amount of digital tokens to participate in block validation.

However, PoS introduces a potential risk of centralization: users holding a large number of tokens may exert disproportionate influence over the network, compromising decentralization.

2.2. Scalability and Layer 2

To improve transaction throughput and overall scalability, several Layer 2 solutions have been developed. These solutions operate as external integrations built on top of the base layer and help overcome performance limitations while preserving the security guarantees of the underlying blockchain.

Although different approaches exist, including State Channels, Plasma and Sidechains, this document focuses exclusively on Rollups, a Layer 2 scalability solution designed to increase blockchain throughput and reduce transaction costs while preserving the security of Layer 1. Rollups have become the most widely adopted Layer 2 strategy, particularly for public blockchains, most notably Ethereum.

2.2.1 Rollups

A Rollup works by executing transactions outside the main blockchain, while still relying on it for data availability and final verification. Instead of having every node on Layer 1 process every transaction, the Rollup aggregates a large number of transactions into a single batch, computes the resulting new state, and submits only the essential data back to the Layer 1 chain. This drastically reduces the amount of computation and storage required on the base layer, which in turn increases throughput and lowers fees. Even though execution happens off chain, Layer 1 blockchain is responsible for verifying correctness through proofs or dispute mechanisms, ensuring that all state transitions produced by the Rollup are valid.

Overall, Rollups represent an effective compromise between scalability, transaction costs and security, and they are rapidly becoming one of the key technologies for improving blockchain performance.

There are two primary types of Rollups:

- **Optimistic Rollups:** Transactions are assumed to be valid by default and processed off chain. If a participant detects an invalid transaction, they can submit a *fraud proof* on Layer 1. The dispute is resolved using Layer 1 verification. This approach significantly reduces processing time and costs, although it introduces a waiting period (challenge window) for potential disputes.

- **Zero Knowledge Rollups (ZK Rollups):** Transactions are processed off chain, but for each batch, a cryptographic *validity proof* is generated and published on Layer 1, which then verifies the proof to ensure it was done correctly. This proof guarantees that all transactions are correct without requiring a challenge period. ZK Rollups offer faster confirmations than Optimistic Rollups, but generating the proofs can require substantial computational resources.

These descriptions provide an overview of Rollups. In the following sections, we will explore each type in more detail, examining their mechanisms, advantages, and limitations.

3. Optimistic Rollups

3.1. Operating Principle

Optimistic Rollups are a scaling approach that involves moving transaction processing and state storage off-chain, outside the main network. Transactions are executed externally, but their related data is still recorded on the primary and official blockchain network where real transactions with economic value occur.

The system functions through operators who group multiple off-chain transactions into large batches, which are later submitted to the blockchain. This approach allows fixed costs to be distributed across many transactions within each batch, which results in reduced fees for users. Additionally, Optimistic Rollups use data compression techniques to minimize the amount of information published on-chain.

The term “optimistic” refers to the fact that these solutions assume that off-chain transactions are valid, and don’t immediately require the publishing of cryptographic proofs of correctness. Instead, after a batch is submitted to the blockchain, a time window called the challenge period (which lasts roughly seven days for Ethereum) opens, during which anyone may dispute the results by providing proof of fraud.

If the challenge succeeds, the disputed transactions are replayed and the rollup state is udpdated accordingly. In this case, the sequencer (the operator who included the incorrect transaction) incurs in an economic penalty.

On the other hand, if the batch is not challenged before the challenge period ends, it is considered valid and permanently accepted by the blockchain. It’s important to note that other rollup blocks may be built on top of a batch that has not yet been confirmed, however, if invalid transactions are later proven, all resulting state changes are retroactively reverted.

3.1.1 Transaction Execution

Users submit their transactions to validators, which are nodes that collect transactions, compress underlying data, and publish blocks to the chain.

Any node may become a validator, but doing so requires depositing a bond, similar to a Proof-of-Stake. The bond acts as financial collateral and may be slashed (partially or fully confiscated) if a validator either publishes an invalid block or builds on top of a previously invalidated block (even if the new block itself is correct). This penalty system promotes honest behavior.

Other validators in the Rollup chain locally execute the same transactions and compare their resulting state with the operator’s proposed state, and if they find any discrepancies, they should initiate a challenge procedure.

3.1.2 Publishing Blocks to the Blockchain

After collecting transactions and building a block, the validator submits the block to the main chain as either calldata or blobs.

Calldata is a non-persistent, immutable memory area of a smart contract. Although it remains stored in blockchain history, it is not part of the blockchain’s state. This means calldata is cheaper for storing data than writing directly to state, and it significantly reduces fees for users.

In Rollup systems, calldata is used to send compressed transaction data to the on-chain contract. The operator adds a new batch by invoking a specific Rollup contract function and passing the compressed data via calldata.

Like calldata, blobs are immutable and non-persistent, but they are removed from network history after about 18 days. This solution further reduces costs and improves scalability.

3.1.3 State Commitments

In Optimistic Rollups, the chain state is organized in a Merkle tree called the state tree. The root of this tree (the state root) represents the rollup's latest state and is stored as a hash inside the on-chain contract.

With every state transition, the validator computes a new state root, which replaces the previous one if it matches the value recorded in the contract. When a new batch is published, the validator must provide both the old and new state roots, and if the old root matches the one stored on-chain, it is replaced by the new one.

In addition to the state root, the validator must also commit to the Merkle root of the transaction batch, enabling anyone to prove the inclusion of a single transaction through a Layer 1 Merkle proof.

The Rollup contract immediately accepts new roots submitted by operators, but may later invalidate incorrect ones and restore the correct chain state.

3.1.4 Proving Fraud

As we previously established, Optimistic Rollups allow validators to publish blocks without providing proof of their validity, and a time window is defined during which anyone may challenge a state transition. Until then, rollup blocks are called assertions, as they may be disputed.

There are two methods of proving the vailidity of an assertion once it is challenged. The first is single-round proof, in which the disputed transactions are replayed through a verification contract in Layer 1. The second involves multiple rounds: dividing the dispute avoids full transaction re-execution on L1 which results in a reduction of costs and on-chain data requirements.

The transition from single-round to multi-round systems represents a fundamental improvement, as it maintains the same security guarantees while lowering costs and improving efficiency.

3.2. Pros and Cons

Table 1: Advantages and disadvantages of Optimistic Rollups

Advantages	*Disadvantages*
Significant scalability improvements without sacrificing security or trustlessness	Transaction finality delays due to potential fraud challenges
Transaction data is stored on Layer 1, improving transparency, security, censorship resistance, and decentralization	Centralized rollup validators may influence transaction ordering
Fraud proving allows honest minorities to protect the chain, and fraud proof computation is accessible to regular L2 nodes	If no honest nodes exist, a malicious operator may steal funds by publishing invalid blocks and invalid state commitments
Anyone can advance the chain by executing transactions and publishing assertions	Users must wait through the challenge period before withdrawing funds
Compatible with EVM and Solidity, allowing developers to deploy native Ethereum smart contracts or reuse existing tools to build new dApps	Rollups must publish all transaction data on-chain, increasing fees

4. Zero-Knowledge Rollups

4.1. Operating Principles

Zero-Knowledge Rollups (ZK-rollups) aggregate transactions into batches executed *off-chain*, thereby reducing the amount of data that must be posted on the main blockchain (Layer 1, L1). Unlike *Optimistic Rollups*, which rely on dispute mechanisms (*fraud proofs*) and require a challenge period, ZK-rollups employ cryptographic validity proofs to attest to the correctness of state transitions.

The overall process consists of three main phases:

1. Transactions are collected and executed *off-chain*, then compressed to minimize publication costs.

2. The operator (sequencer) submits a state commitment to the *mainnet* – typically the new *state root* – accompanied by the *validity proof.* At the same time, the relevant *calldata* (or

its cryptographic commitment) is published to guarantee *data availability.*

3. The rollup's smart contract on L1 verifies the proof and securely updates the on-chain state commitment.

4.2. Transaction Execution

Users of a ZK-rollup sign their transactions and send them to L2 operators for processing and inclusion in a forthcoming batch. In many implementations, the operator is a centralized entity, referred to as the *sequencer*, which is solely responsible for executing transactions, constructing L2 blocks, and submitting batches to the rollup contract on L1. Under this model, only the *sequencer* can advance the L2 state.

Other ZK-rollups adopt validator-based systems using Proof of Stake. Prospective operators must lock funds in the rollup contract, and their probability of being selected to produce the next batch is proportional to their stake. Malicious behavior may result in slashing, providing strong economic incentives for honest participation.

4.3. Publishing ZK-Rollup Data on Ethereum

In ZK-rollups, aggregated transaction data is published on Ethereum as *calldata*, ensuring *data availability* and allowing anyone to reconstruct the L2 state independently of the operator. To reduce gas costs, the published data often consists only of a cryptographic commitment—for example, the *batch root*—rather than the full transaction set.

4.4. State Commitments

The state of a ZK-rollup, including all L2 accounts and balances, is represented using a *Merkle tree*. The tree's root, known as the *state root*, is stored in the on-chain rollup contract, enabling secure tracking of state evolution. Each transaction batch results in a new state, represented by a new *state root.*

In addition, the operator computes the *batch root*, the Merkle root of the transactions included in the batch. This batch root enables users to prove the inclusion of individual transactions through *Merkle proofs*, without requiring the download of the entire L2 block.

4.5. Validity Proofs

To ensure the correctness of all state updates, each batch is accompanied by a *validity proof*, a cryptographic proof verified by the on-chain smart contract before the corresponding *state root* becomes canonical. The validity proof guarantees that all transactions in the batch are valid and consistent with the previous state, without requiring L1 to re-execute them.

This mechanism allows immediate state finalization and enables users to withdraw funds without challenge periods or delays, while maintaining a trustless security model.

5. Advantages and Disadvantages

Table 2 summarizes the main advantages and disadvantages of Zero-Knowledge Rollups as a scalability solution.

Table 2: Advantages and disadvantages of ZK-rollups

Advantages	*Disadvantages*
Validity proofs ensure correctness of all off-chain transactions, preventing invalid state transitions	Computing and verifying validity proofs is resource-intensive, increasing operational costs
Fast transaction finality: state updates are confirmed as soon as the proof is verified on L1	Generating validity proofs often requires specialized hardware, potentially contributing to centralization
Security relies on trustless cryptographic guarantees rather than on the honesty of incentivized participants, as in Optimistic Rollups	Centralized sequencers may influence transaction ordering (MEV risks)
On-chain data availability guarantees security, decentralization, and censorship resistance	High hardware requirements may reduce the number of participants capable of producing blocks, increasing the risk of operator-level stagnation or censorship
Greater capital efficiency and the possibility of withdrawing funds from L2 without delay	Some proving systems (e.g., ZK-SNARKs) require a *trusted setup* which, if compromised, may endanger rollup security

Advantages	*Disadvantages*
Users do not need to validate the chain to protect their funds, and the system does not rely on liveness assumptions	
Superior data compression reduces calldata publication costs and lowers transaction fees	

6. Comparison Optimistic vs ZK Rollups

In this chapter, we compare the main characteristics of the two primary rollup models: Optimistic Rollups and ZK-Rollups.

6.1. Transaction Validation

Optimistic Rollups. Transactions are considered valid by default. However, there is a challenge period during which anyone can contest the correctness of a batch of transactions by providing a fraud proof. If the batch is found to be incorrect, the state is corrected.

ZK-Rollups. Transactions are not automatically valid. Each batch must be accompanied by a cryptographic validity proof, which guarantees the correctness of the transactions before the state is updated on L1. Once the proof is verified, transactions become final immediately.

6.2. Withdrawal of Funds

Optimistic Rollups. Withdrawals are delayed to allow for the challenge phase; users must wait until the end of the challenge period before they can withdraw their funds.

ZK-Rollups. Funds can be withdrawn as soon as the contract verifies the validity proof, ensuring immediate finality and a better user experience.

6.3. Validity Proofs

Optimistic Rollups. No validity proofs are published on-chain; security relies on the ability to contest incorrect batches through fraud proofs.

ZK-Rollups. Validity proofs are published on-chain together with the summary of the updated state. This allows the contract to verify the correctness of batches without having to recompute them.

6.4. Operational Mechanism

Optimistic Rollups

1. Aggregate multiple transactions into batches and send compressed data to the blockchain.
2. Transactions are executed off-chain, while the data is published on the mainnet.
3. A challenge period begins; uncontested batches are accepted as valid, while contested ones are corrected if necessary.

ZK-Rollups

1. Aggregate transactions into batches.
2. The operator submits a summary of the state changes along with the validity proof.
3. Once the proof is verified by the contract, the transactions are finalized and the updated state is considered valid.

7. Conclusion

The analysis of Optimistic Rollups and Zero-Knowledge Rollups has highlighted how both solutions represent fundamental tools for overcoming the scalability limits of Ethereum and, more generally, of public blockchains.

Optimistic Rollups, based on dispute mechanisms (fraud proofs), offer a high level of compatibility with the EVM and guarantee security through cryptoeconomic incentives, but entail longer finalization times due to the challenge periods.

ZK-Rollups, on the other hand, thanks to the use of cryptographic proofs (validity proofs), ensure immediate transaction finality and greater efficiency in resource management, although they require more specialized hardware and may involve a higher risk of operator centralization.

In general, both solutions contribute to making blockchains more scalable, reducing transaction costs and times without compromising the security and decentralization guaranteed by Layer 1. The choice between Optimistic and ZK-Rollups depends on specific needs regarding scalability, speed, and computational complexity, as well as the application context. In the future, the evolution of Rollups, integration with sharding techniques, and improvements in validity proofs could make these solutions even more performant and accessible, paving the way for a new generation of decentralized applications with high transaction throughput and for large-scale adoption of blockchain in real-world contexts.

Bibliography

[1] Dvorchuk, D., Shpinareva, I. (2025) Analysis of Blockchain Technology. https://www.researchgate.net/publication/392603297_Analysis_of_Blockchain-Technology

[2] Thibault, L. T., Sarry, T., & Hafid, A. S. (2022). Blockchain scaling using rollups: A comprehensive survey. IEEE Access, 10, 93039-93054.

[3] Vashchuk, O., & Shuwar, R. (2018). Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake. Electronics and information technologies (09).

[4] Binance Academy, *What is the blockchain trilemma?*, 2019. https://www.binance.com/en/academy/articles/what-is-the-blockchain-trilemma

[5] Binance Academy, *Optimistic vs. Zero-Knowledge Rollup: What's the difference*, 2023. https://academy.binance.com/en/articles/optimistic-vs-zero-knowledge-rollups-what-s-the-

[6] Binance Academy, *Sharding.* https://academy.binance.com/en/glossary/sharding

[7] Ethereum, *Proof-of-stake vs proof-of-work*, 2024. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/pos-vs-pow/

[8] Ethereum, *Optimistic Rollups*, 2025. https://ethereum.org/nb/developers/docs/scaling/optimistic-rollups/

[9] Ethereum, *Scaling*, 2025. https://ethereum.org/en/developers/docs/scaling

[10] Guardarian, *Blockchain layers explained*, 2023. https://guardarian.com/blog/blockchain-layers-explained-l1-l2-l3/